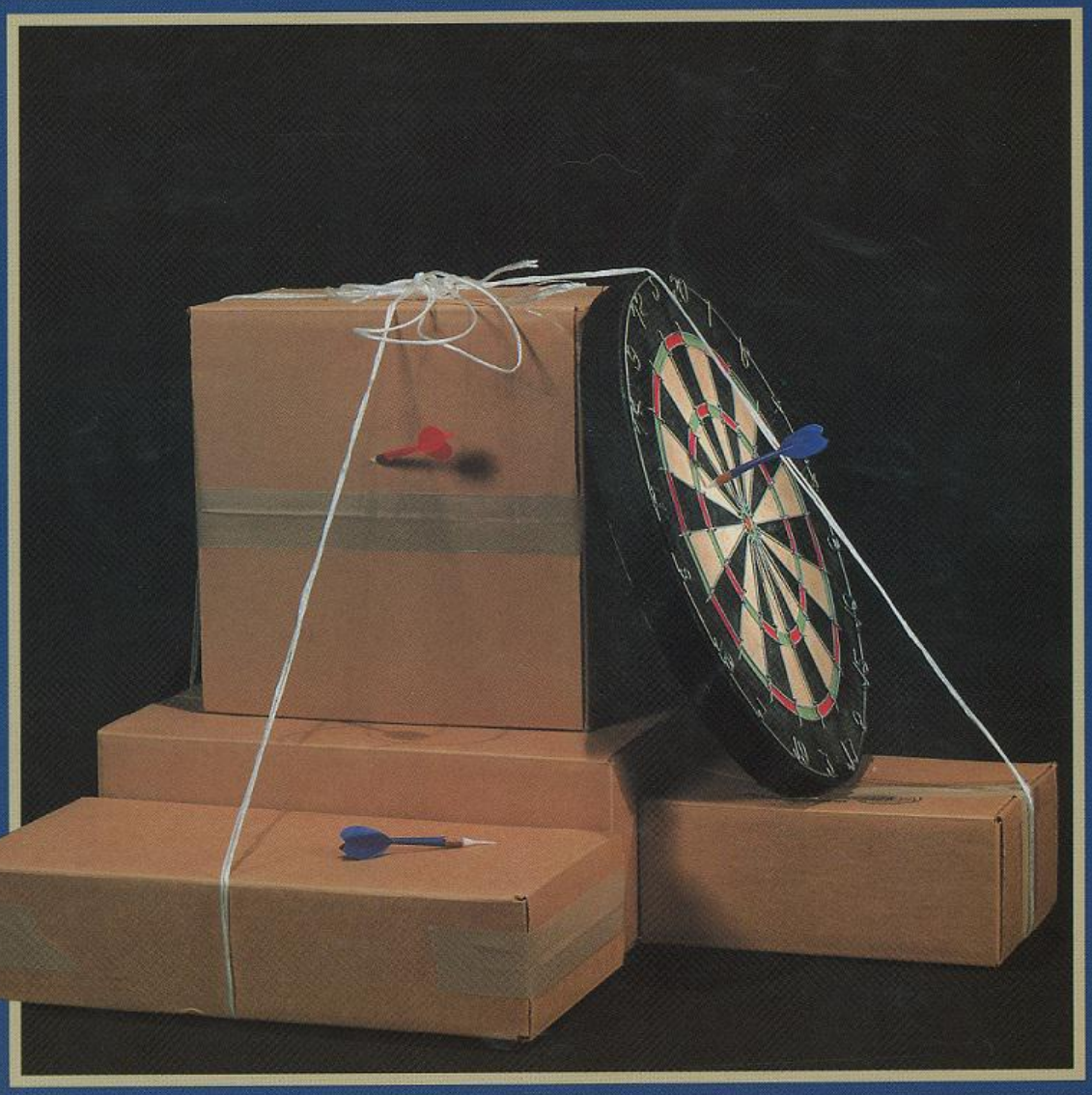
HOWARD W. SAMS & COMPANY

22590

IBM® PC Advanced Troubleshooting & Repair

Robert C. Brenner



IBM® PC Advanced Troubleshooting & Repair

IBM® PC Advanced Troubleshooting & Repair

Robert C. Brenner

HOWARD W. SAMS & COMPANY
A Division of Macmillan, Inc.
4300 West 62nd Street
Indianapolis, Indiana 46268 USA

WARNING: Opening or otherwise modifying the IBM PC may void any manufacturer's warranty on the product.

WARNING: Dangerous voltages and currents are found in the IBM PC power supply. Only trained technicians should troubleshoot in or around power supplies.

WARNING: Dangerous voltages and currents are found in the display monitor used with the IBM PC. Only trained technicians should troubleshoot in or around video display terminals.

FIRST EDITION
FIRST PRINTING—1987

International standard book Number: 0-672-22590-5
Library of Congress Catalog Card Number: 87-62223

Acquisitions Editor: *Greg Michael*
Editor: *C. Herbert Feltner*
Illustrators: *Don Clemons & Sally Copenhaver*
Word Processor: *Kim Clark*
Cover Graphic: *James R. Starnes*
Compositor: *BMEP, Inc., Zionsville, IN*

Trademark Acknowledgements

All terms mentioned in this book that are known to be trademarks or service marks are listed below. In addition, terms suspected of being trademarks or service marks have been appropriately capitalized. Howard W. Sams & Company cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

IBM is a registered trademark of International Business Machines Corporation.

COMPUTERFACTS is a trademark of Howard W. Sams & Company

Printed in the United States of America

This book is dedicated to my seventh-grade teacher, Lillian Siefert,
for the encouragement and support that motivated me to seek a writing career.
Her smiles from heaven continue to inspire me through long,
arduous hours of technical research and writing.

Contents

Acknowledgement

My personal thanks to my son, Dan, for his technical support in producing this document, and to Ed Roxburgh for his expertise in producing the technical illustrations used throughout this manual.

Introduction

Why a book on a machine that was originally introduced in 1981?

When IBM unveiled the IBM Personal Computer (PC) during the summer of 1981, it was a dream come true for many aspiring microcomputer manufacturers. The decision by "Big Blue" to enter the microcomputer market made these machines credible and acceptable by businesses everywhere. The legitimizing of microcomputers in the office environment by the introduction of the IBM PC produced an impact on business that has been felt around the globe. Almost overnight, Fortune 500 companies sat up and took a serious look at uses of the microcomputer in their environment. And applications were discovered that seemed impossible (at least for a desktop computer).

Suddenly the "in thing" was to own a PC. Sales of IBM PCs skyrocketed. So did sales of non-IBM microcomputers. And the personal computer revolution was on.

IBM's professional approach to design, manufacturing, marketing, sales, and support ensured that the PC would have a long and useful life. Six years after the first PC was sold, over 3 million PCs are still in use.

Documentation is an important area of support for the success of any new product, and over a dozen books have been published on the IBM PC. Most books cover the use of the machine from a software operation perspective. To meet the needs of understanding the hardware of the PC, IBM developed the *Technical Reference* manual and the *Hardware Maintenance and Service* manual.

While these were (and still are) useful reference documents, more information was requested by the consumer. Howard W. Sams & Company published the *IBM PC Troubleshooting & Repair Guide* in early

1986. This book was closely followed by the IBM Model 5150 COMPUTERFACTS™. The former book served as the bridge between the owner's manual that came with the machine and the service center schematics found in the COMPUTERFACTS. Yet, this still did not completely answer the needs of the user. What was needed was an intermediate to advanced book that described the detail found in the IBM PC COMPUTERFACTS for computer service technicians, service center technicians, advanced hobbyists, and educational institutions.

Meanwhile, schools and universities teaching microcomputer repair needed a text that would guide the student through troubleshooting and repair based on a well-known and widely accepted machine–the IBM PC.

This book was developed to meet these needs. It is intended to complement both the Sams COMPUTERFACT and the *IBM PC Troubleshooting & Repair Guide*. This advanced technical book complements the COMPUTERFACT service data with descriptive text and expanded troubleshooting and repair circuit explanations. It is written to complete the documentation requirements of the using public and all the service centers and repair shops that troubleshoot and repair this marvelous machine. It can be used as a text for a course in microcomputer troubleshooting and repair. Its intent is to make better repair technicians out of us all.

Three types of service center technicians can be found in the industry today:

- Mechanics
- Bulldozers
- Professionals

The "mechanic" examines the circuit boards looking for a visual cause for the problem. This person performs only the preliminary steps in troubleshooting and then seeks a simple way to fix the failure. This is the type of individual who will bang the side of a chassis to see if this corrects the problem, or who does a 3-foot drop to see if that will produce a failure. It often does, although it typically causes a new failure without changing the original problem. This is the same person who persistently sprays the circuit board components with cool spray, then applies the heat gun which heats chips until the failure changes or goes away. This person is totally unaware that the operating life of most of the components that have been frozen and then overheated has been significantly reduced.

The "bulldozer" observes the symptoms, isolates the problem to a failing subsystem, and then replaces every related part in that subsystem. This is certainly not a professional approach to troubleshooting and repair.

The "professional" follows a careful, methodical process to identify and isolate a problem. Much like a detective, this person reads all the available technical documentation, has the necessary information in view, and uses all the right equipment and tools to recognize the clues and follow the indications that lead to a failure that can be "surgically" corrected.

This book was written to help you become known by your actions–to help you become a professional technician.

The book begins with a systems overview of the IBM PC. Each major part of the machine is described in general terms.

Chapter 2 is a detailed description of the operation of the PC system. Many schematic subsets of the IBM PC COMPUTERFACTS are included to give you an in-depth understanding of the signals and circuitry associated with each major signal, address, clock, and data. The book is intended to supplement the Sams COMPUTERFACTS so you will get both the broad macro view and close-up microscopic view of the circuitry.

In Chapter 3, you are guided through the techniques and tricks used by service technicians to troubleshoot and repair this machine. This includes cold troubleshooting, when the technician has no idea of the problem, which instructs you in ways to isolate the problem to a specific area. Use of the tools of the trade is also covered in this chapter.

Preliminary service checks are addressed in Chapter 4. It's here that the troubleshooter confirms that the problem is not operator error or a software malfunction and isolates a symptom to a particular area of circuitry.

Once the area of the problem has been isolated, Chapter 5 guides the technician through the detailed circuit troubleshooting analysis that leads to identification of the failed part. This chapter covers problems in the power supply, the system board, the monochrome monitor/printer adapter board, the color graphics monitor adapter board, the keyboard, and the disk drives and disk drive adapter board.

Comprehensive appendices cover disassembly and reassembly, adjustments to the system and power boards, notes regarding the schematics, switch and jumper settings, and a list of the safety precautions and warnings found in the text.

Troubleshooting can be very frustrating if you are left to struggle through the process by yourself without a good guide. This book provides the techniques for quick and easy troubleshooting and repair. It is the first in a series of advanced troubleshooting and repair manuals that Howard W. Sams & Company is producing. Its existence makes all our jobs easier and much, much clearer.

1

System Overview

Before attempting any troubleshooting analysis on a system, you must first understand the system configuration and how the system works. In this chapter, you will approach the IBM PC from the systems level and become acquainted with each functional element in the microcomputer. From your introductory computer instruction, you learned that every computer system has five basic parts (Fig. 1-1):

- An arithmetic logic unit
- A control unit
- A memory unit
- An input unit
- An output unit

The arithmetic logic unit (ALU) does the adding, subtracting, multiplying, dividing, comparing, and logic operations. A control unit regulates the operation of the complete machine. It fetches and interprets instructions, and it causes certain parts of the circuitry to respond according to those instructions. The ALU and control unit can be considered the nerve elements of the computer's brain. Important to the computer is adequate power and a good system clock.

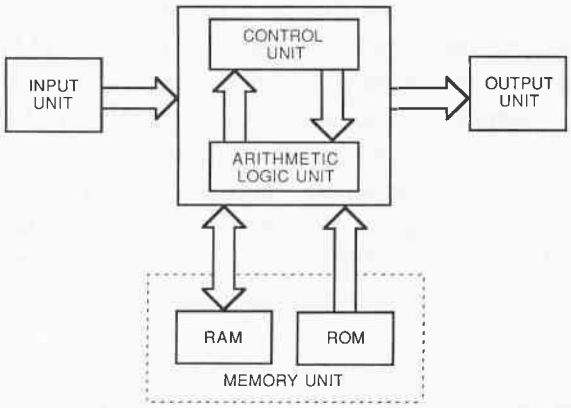


Fig. 1-1. Basic parts of the IBM PC computer.

The memory unit is the remembering part of the machine. It stores programs, data, calculations, and results. Two types of memory are included in the PC: temporary memory called random-access memory (RAM) and permanent memory called read-only memory (ROM). The ROM chips are permanently programmed or written into during manufacture with computer instructions and special data. Since one of the

IBM ROMs has an operating system program permanently stored in it (hardware), we call it "firmware."

RAM is sometimes called "main memory." Information stored in RAM exists only as long as power is applied to the computer. When power is removed, the programs and data stored in the RAM are lost unless transferred (copied) to external permanent memory. External memory can be attached to the PC by connecting floppy disk drives or hard disk drives.

Communication with the computer occurs through the input and output units. These person-machine interfaces are called "peripherals." A peripheral can be just about any type of device connected to the basic PC.

An input unit allows commands, programs, and data to enter the computer–It's how the user "talks" to the computer. Keyboards, joysticks, game paddles, graphics tablets, light pens, microphones, and analog-to-digital converters (ADCs) are examples of input devices.

The computer communicates with our environment and us via an output unit. Monochrome and color displays, printers, plotters, a speaker, and digital-to-analog converters (DACs) are examples of output devices.

Some devices are used for both input and output. These include the mass storage devices (the floppy disk drives, the hard disk drives, and archival storage tape systems) and MOdulator-DEModulators (MODEMs) which enable computer to computer communications over long distances.

The arithmetic logic unit and control unit can be combined into a single integrated circuit called a "central processing unit" as shown in Fig. 1-2. Microprocessors are also called central processing units (CPUs) because they can be designed to do the same functions as the central processing unit in a large computer. In the IBM PC, an 8088 integrated circuit (IC) is the systems central processing unit. It accesses memory (fetches an instruction), interprets what the instruction means, and does the actions required by the instruction, and then fetches the next instruction in the program and repeats the sequence. This sequential process was described by mathematician John Von Neumann in the late 1940s. Today, every desktop or laptop personal computer operates as a sequential "Von Neumann" machine. In the future, microcomputer systems will include multiple processors and operate on many instructions at the same time in a parallel "non Von Neumann" architecture.

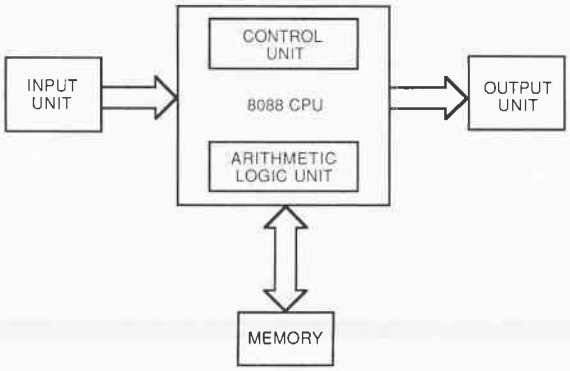


Fig. 1-2. The control unit and arithmetic unit together form the central processing unit CPU.

THE BASIC IBM PC INTEGRATED INTO A SYSTEM

Taken as a whole, an IBM PC system could look like the configuration shown in Fig. 1-3.

A typical PC system is comprised of the system unit with keyboard, a display unit, two disk drives, and a printer. The next few pages will describe each of these units.

System Unit

The system unit shown in Fig. 1-4 is the main component of the computer. It houses the system board (motherboard) with its expansion slots for external interface peripheral devices, the switching power supply, and two standard-height, double-density disk drives. On the right side, near the back of the system unit, is the on-off power switch.

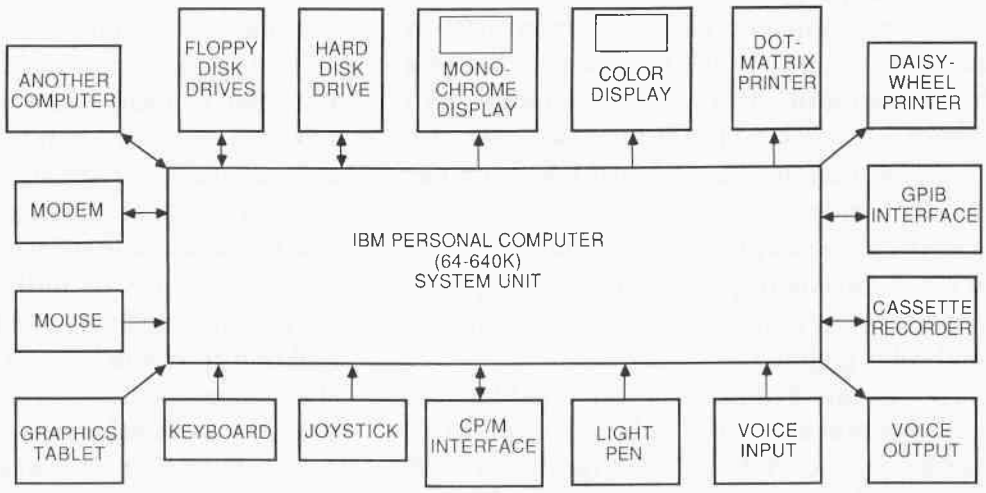


Fig. 1-3. Example of an IBM PC system.

Fig. 1-4. The system unit sits behind a detachable keyboard.

Keyboard

The detachable keyboard has 83 keys that can generate all 128 American Standard Code for Information Interchange (ASCII) characters. It can also generate special symbols and graphic shapes. In all, the keyboard can cause the machine to generate and display (or print) 256 characters, shapes, or symbols.

The numeric keypad on the right side of the keyboard has some keys that double as cursor-control keys. Ten programmable function keys are mounted on the left side of the keyboard to execute specific programs or to initiate special software routines. The functions can be programmed by the software designer. Many special keys are provided including Shift, Up-Down-Right-Left arrow keys, Caps Lock, Number Lock, Scroll Lock, Backspace, Enter (or Return), Home, Page Down, Page Up, End, Delete, Insert, Print Screen, Tab, Control, and Alternate. The functions of these keys and key combinations are described in the *IBM Guide to Operations* manual.

All 83 keys have automatic repeat and a 10-character type-ahead buffer to let you type at rapid speed without getting ahead of the computer processing of each key stroke.

Each of the 83 concave keys have a tactile feel with an audible click to provide positive feedback that key action has been completed. Inside the keyboard are electronic circuits that enhance key operation and permit keys to be redefined for increased programming flexibility.

On the bottom side of the keyboard are two plastic feet that allow the keyboard to be tilted in two positions for best typing comfort. A plastic ridge above the top row of keys can hold a book or report between the keyboard and the display screen. It can also hold templates for special application software. A 6-foot coiled cable connects the keyboard to the rear of the system unit.

Output Unit

Two output units that make the computer system complete are the display unit and a printer.

Many display units can connect to the IBM including monochrome and color monitors. A display unit connects to the computer at the rear of the system unit. If a radio frequency (RF) modulator is connected to the video adapter card inside the system unit, a standard television can be used for a monitor.

Two video adapters are available for the PC: a monochrome display adapter that supports text, and a color/graphics adapter that supports text and color graphics.

The monochrome adapter enables the system to generate and display 25 rows of 80 characters each in white on black (green on black with the IBM monochrome monitor), black on white (or black on green), blinking, in high intensity, or underlined. This adapter card also has a connection for the IBM dot-matrix printer.

Color is available using the color-graphics adapter card. This interface provides options for two types of text (25 rows of 40 characters, or 25 rows of 80 characters) and three types of graphics (low resolution, medium resolution, and high resolution). Only medium resolution and high resolution are supported by the ROM. The 6845 CRT controller IC on the adapter board must be directly addressed by custom software to enable low-resolution graphics.

Low-resolution graphics refers to 100 rows of 160 pixels (picture elements) or dots each, in any of 16 standard colors. Medium-resolution graphics can produce 200 rows of 320 pixels per row in any of four colors. Additional colors can be generated when dots of different colors are juxtaposed. High-resolution graphics produces 200 rows of 640 pixels per row in black and white (or black and green). Text can be positioned within graphic shapes enabling window operations.

Many types of printers can connect and function with the PC. Both dot-matrix and full character printers are commonly connected to this machine. The recent introduction of consumer laser printers has given very high quality hard copy output capability to PC users. All of these interconnections are achieved using a special expansion slot adapter board as the interface. Some users have both types of printers connected at the same time–dot-matrix for drafts and working copies, and letter quality for documents.

A final output device that is part of the PC is a small (2-inch) 8-ohm speaker mounted at the left side of the system board inside the system unit chassis. This device can produce the familiar beep, arcade, and music sounds. It can also produce crude speech.

Two types of bidirectional data storage devices are used with the IBM PC: tape cassette and minifloppy drives. While cassette storage is slow, magnetic tapes provide an excellent way to provide archival and storage for the large amounts of data that are generated every day. Many more files or pages of information can be stored on a good audio cassette tape than can be stored on a 5$^1/_2$ inch floppy disk. In fact, one form of tape archival storage uses video tape for long term computer-generated data.

The typical system uses minifloppy disks that operate in a disk drive unit as the mass storage medium for the PC. Up to six disk drives can be connected to the PC using non-IBM hardware.

Each 40-track disk can be single- or double-sided, double-density depending on the drive used. The disks are magnetically sectored during formatting into 512-byte sectors providing 163,840 bytes of storage for single-sided, double-density disks (184,320 bytes with PC-DOS 2.x) or 327,680 bytes of storage for double-sided, double-density disk drives systems (368,640 bytes for PC-DOS 2.x).

Connections

Figure 1-5 shows the connections on the rear of the PC. A female connector provides power to an external monochrome display. The male connector to the right is for the power cord. To the right of the round fan air exhaust port is a 5-pin circular connector for the keyboard cable. Next is a 5-pin circular connector for cassette data input/output. The five slots align with five expansion sockets on the system board inside. These slots are for connecting other display outputs, disk drives, plotters, printers, and other peripherals.

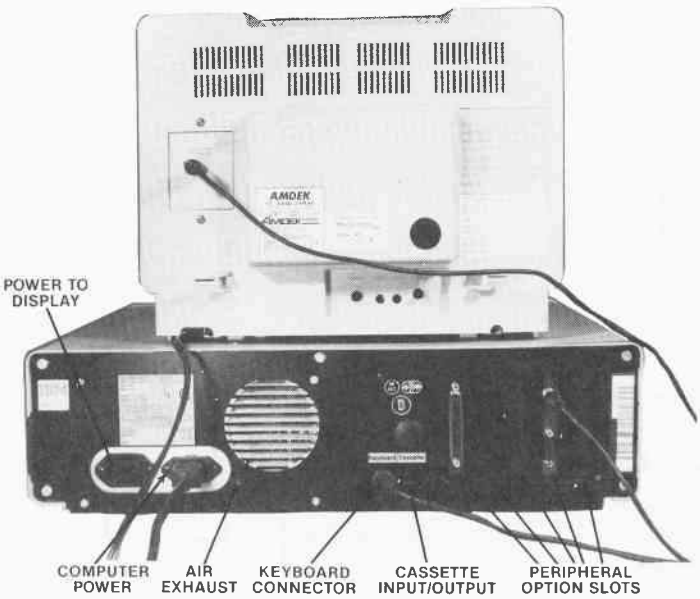


Fig. 1-5. The connections found on the back of the PC system unit.

Internal Components

Figure 1-6 shows the position of subsystems inside the 5.5 inch by 19.6 inch by 16.1 inch system unit. On the left is the system board, or motherboard. At the top of this board are the five expansion slots to connect peripheral devices. On the lower left of the system board are the memory chips.

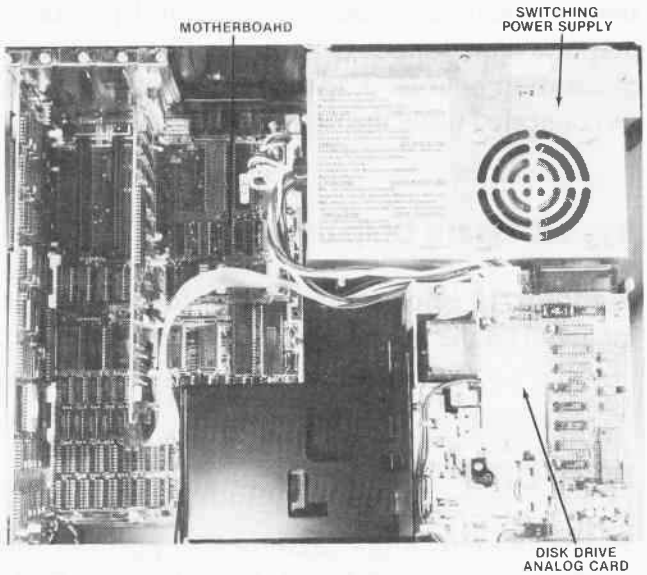


Fig. 1-6. Inside view of system unit

The top right portion of the system unit is the switching power supply. The lower two-thirds of the system unit houses two floppy disk drives.

Figure 1-7 is a block diagram of the IBM PC system. Dashed lines allocate the system elements into the basic parts of any computer.

Each part of this diagram will be covered in detail in the next chapter.

System Board

The most important part of the computer is the CPU and the circuitry surrounding the CPU. On the PC, the CPU and its associated circuitry are mounted on a densely packed printed-circuit board called the system board, or motherboard. Two system boards have been produced for the IBM PC. The board shown in Fig. 1-8 was designed for systems that contained between 16k and 64K of RAM.

A photograph of the system board used in newer 64K to 256K RAM systems is shown in Fig. 1-9. These parameters define the amount of RAM that can be installed in the system board before expansion memories can be used. COMPUTERFACTS pages 27 and 34 provide another view of the newer PC system board. The trace side is shown on CF pages 26 and 35.

As noted, most of the ICs on these boards are mounted in a common direction with pin 1 of each chip facing the same direction. This is important when these boards are repaired because it helps to prevent you from mounting the chip backwards. Chips are marked with the package positioned in the same way. If you inadvertently solder a replacement IC into the board with pin 1 facing the wrong way, you can quickly tell by the upside down lettering and numbering that it's mounted incorrectly.

Each component IC has a corresponding code (U23, U36, and so forth) stamped on the system board. In addition, the chip locations are marked in increasing order from top to bottom and right to left. This is helpful in quickly locating ICs.

In Chapter 2, all system board chips will be discussed in detail. For consistency, only the

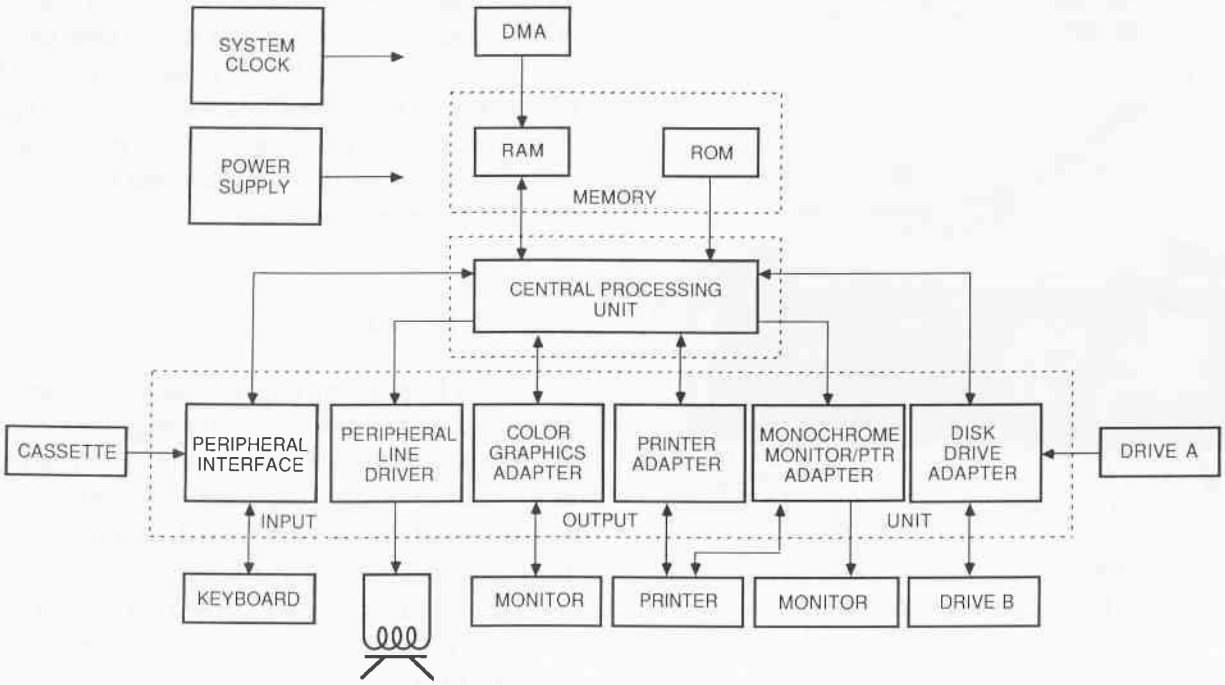


Fig. 1-7. IBM PC system block diagram.

64K-256K board will be described. The 8088 CPU is plainly visible on the lower right side of the board. In all, 99 chips are mounted on the system board. These chips comprise the CPU, memory, timer, controllers, and input/output (I/O).

I/O Interfaces

Five 62-contact expansion slots are mounted at the upper right in Fig. 1-9. These slots connect peripheral devices to the system board. Besides the address and data buses, an extensive selection of control, power, and ground signals available on each slot allow PC system interfacing with many external devices. Each peripheral except the keyboard connects to the system board via an adapter board that plugs into one of the expansion slots. These slots are numbered J1 (slot 1) to J5 (slot 5) from left to right.

Slot 1 is allocated for the disk controller interface board. Up to four disk drives (two internal and two external) can be connected to the PC. Third party hardware can expand the mass storage to six double-sided, double-density drives.

Slot 2 is used for the display adapter card. Slot 3 is allocated to the asynchronous/synchronous communication card. Other interface devices include modems, additional printers, graphics tablets, and voice-recognition and voice-generation boards. Expanded memory can also be connected using the expansion slots.

System Board Chip Layout

Figure 1-10 shows the layout for the chips mounted on the newer IBM PC system board.

8088 CPU

Just to the left of the cassette I/O port (J6) on the far right side of Fig. 1-10 is the 8088 CPU

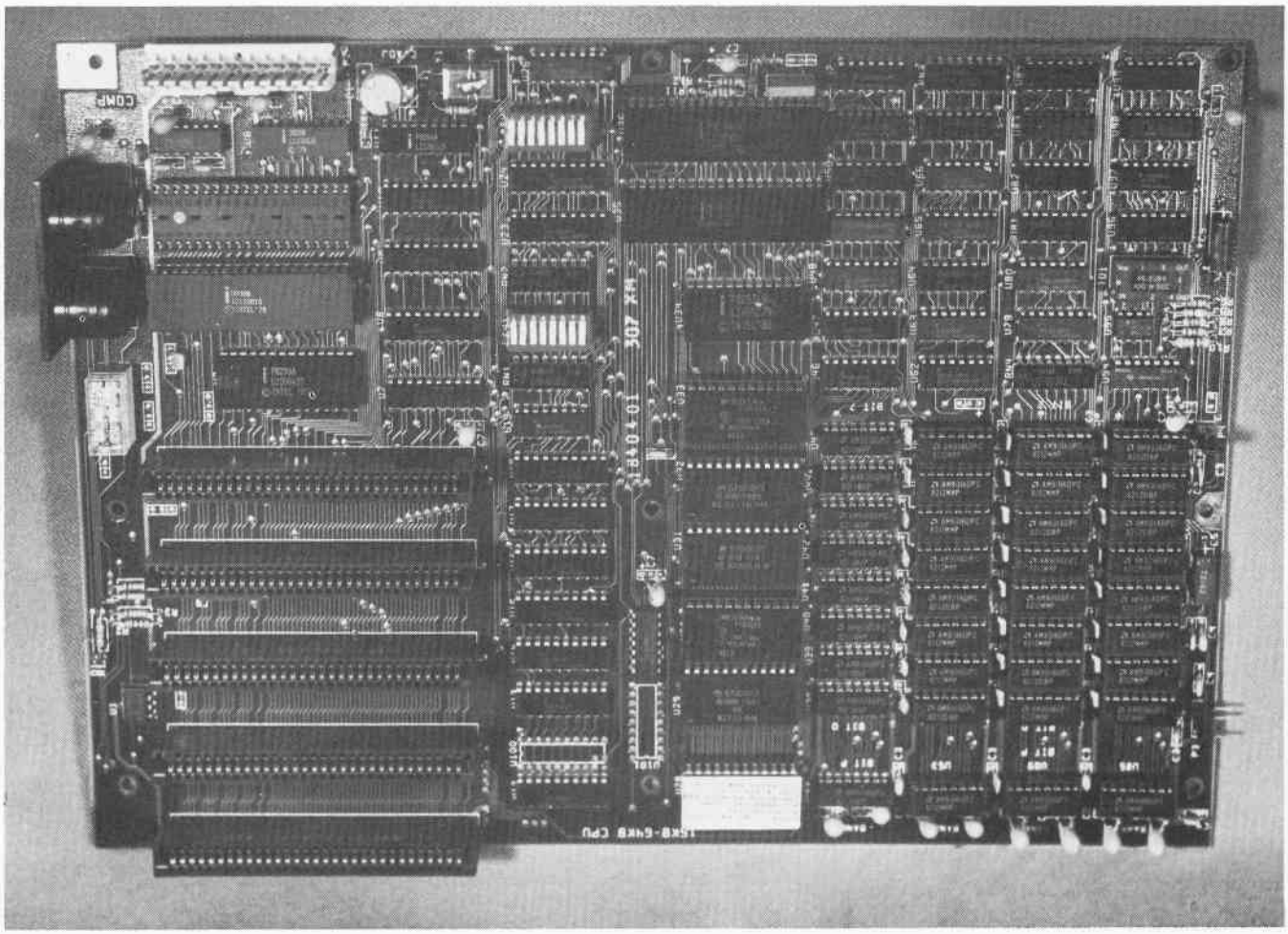


Fig. 1-8. Photo of the 16K-64K IBM PC system board.

(socket U3). A closer view is provided in Fig. 1-11. The 8088 is an Intel designed microprocessor that operates on a 4.77 MHz clock as a 16-bit machine internally, using the same instruction set as the 16-bit Intel 8086 microprocessor, but with an 8-bit data bus. It supports 16-bit operations including multiply and divide and has a 20-bit address bus so it can access over a million memory locations.

Internally, the 8088 handles 16-bit formats so the CPU expands its internal address word to 20 bits at its output using a segmentation scheme. Memory addresses are logically subdivided into special segments of 64K bytes each. Each segment can be allocated to special registers in the 8088. Then bytes within a segment are addressed using a 16-bit offset address which is added to a 16-bit segment address to generate a physical address. This addressing scheme will be described in detail in Chapter 2.

8087 NUMERIC DATA PROCESSOR (COPROCESSOR)

The 8088 can also operate in maximum mode with an optional 8087 numeric data processor functioning as a coprocessor. This configuration greatly increases computational speed. The socket just below U3 is for the 8087 (U4). The 8087 is a high-speed, two-channel I/O controller/coprocessor that extends the 8088 instruction set to include arithmetic and logic operations. It doesn't change the way the system operates, but

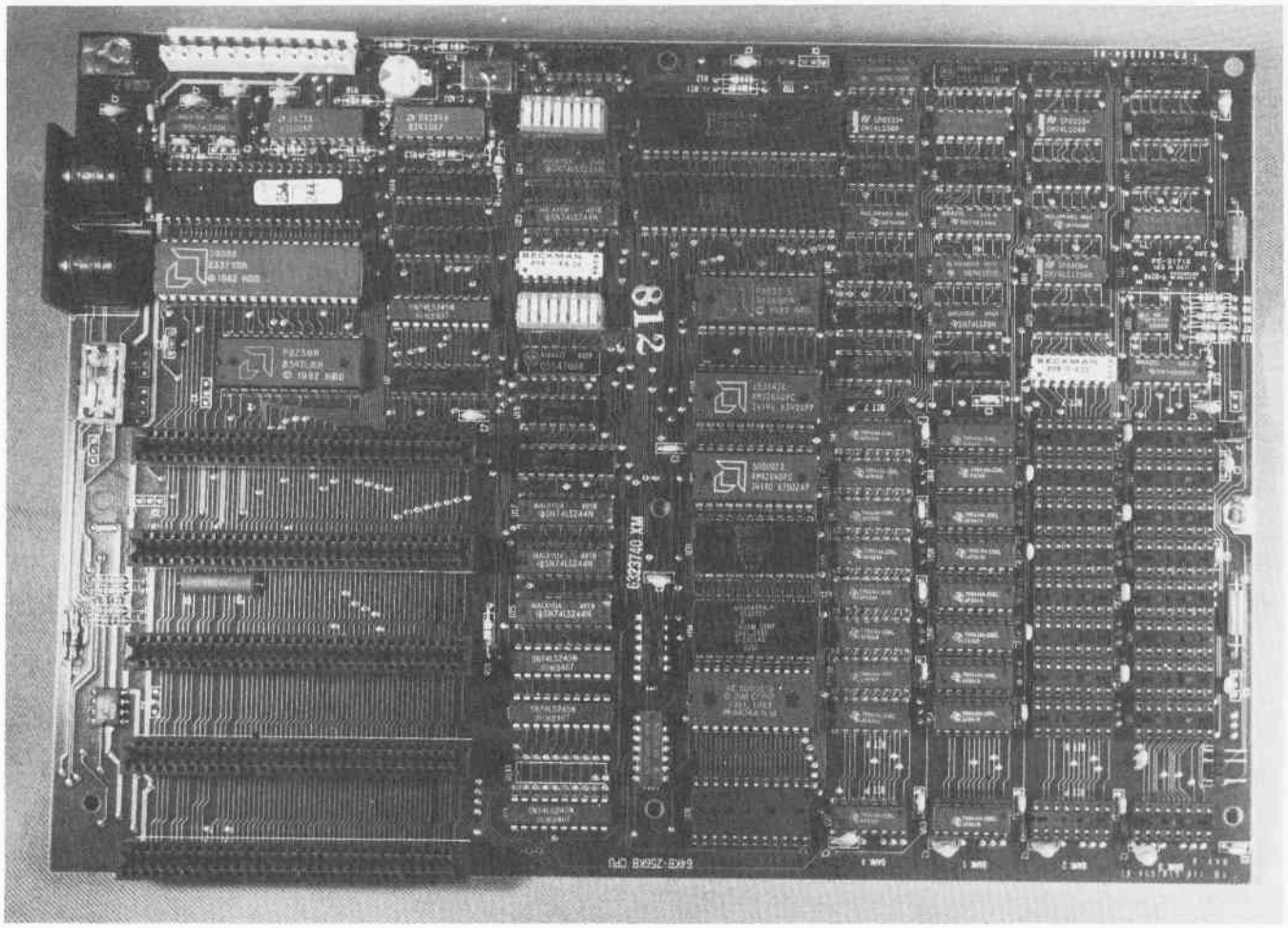


Fig. 1-9. Photo of the 64K-256K IBM PC system board.

it does greatly reduce the time required for certain mathematical functions. This IC can operate on floating point and multiple digit BCD numbers up to 18 digits in length.

The 8087 numeric data processor handles the data transfers between it and the 8088. Permanently stored in the 8087 are micro instructions for add, subtract, multiply, divide, absolute value, arctangent, tangent, square root, and other operations. Its unique "number-crunching" architecture enables calculations 100 times faster than the 8088. When installed, a switch on the system board enables the 8087 CPU to quickly download mathematic operations to its 8088 coprocessor, dramatically reducing execution times of these algorithms.

8259 PROGRAMMABLE INTERRUPT CONTROLLER (PIC)

Just above the 8088 shown in the layout diagram of Fig. 1-11 is the 8259 PIC (U2) that produces special signals that are used for external device communication with the CPU. The PC is an interrupt driven machine in which peripheral devices communicate with the CPU by interrupt signals that cause the CPU to stop what it was doing and service their request. Up to eight external devices can request PIC U2 to produce a CPU interrupt signal. By knowing which device is requesting the interrupt, U2 puts a special code out on the data bus for CPU recognition and action.

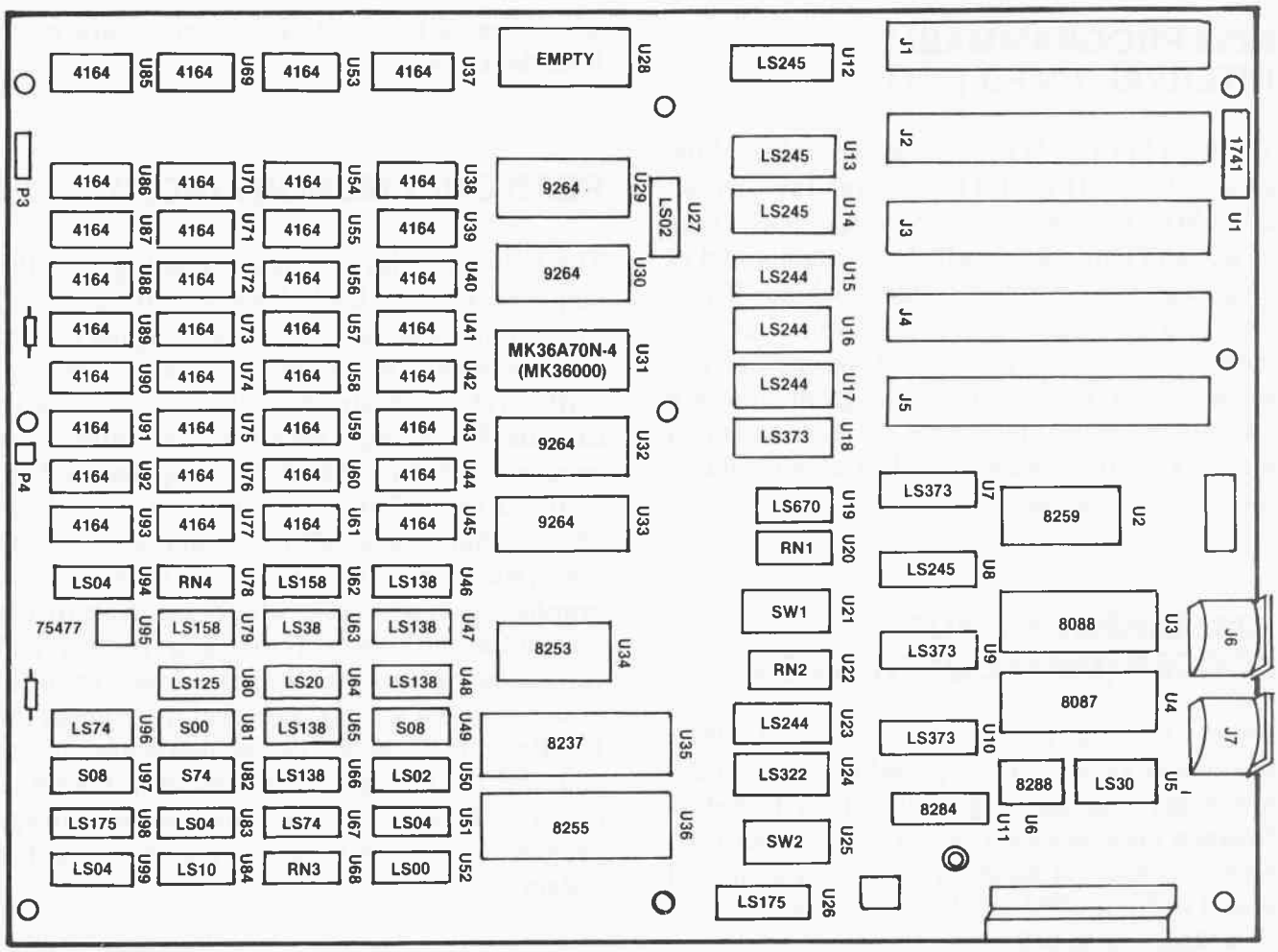


Fig. 1-10. The system board chip layout

8288 CLOCK GENERATOR

In the lower right of Fig. 1-10 is an 8288 clock generator (U11) that receives a power good signal from the power supply and a 14.31818 MHz signal from an attached crystal and produces the *reset* pulse and clock signals to start the CPU operating and awaken the circuitry of the system board into electronic life.

Once powered, this IC continuously produces several clock signals that pulse throughout the PC. A 4.772727 MHz signal is passed to the 8088 CPU and out onto the system board as the system clock. The clock generator also produces a 2.386 MHz signal that is divided by two and used to refresh the dynamic RAM on the system board, and to update the time-of-day internal PC clock. It also is used to activate the speaker.

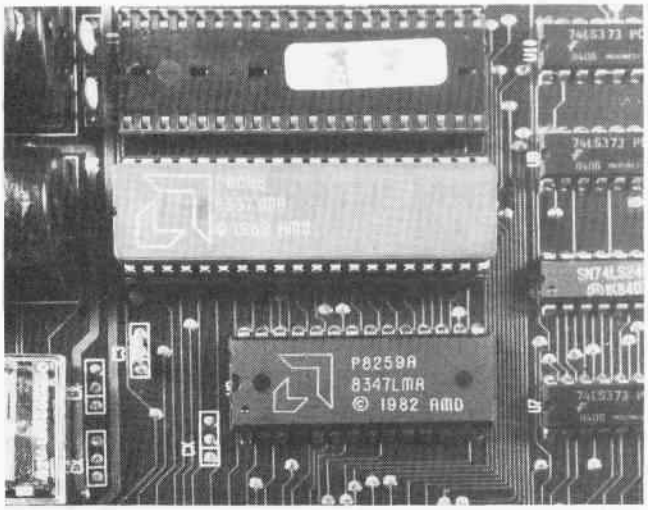


Fig 1-11. The 8088 central processing unit.

8253 PROGRAMMABLE INTERVAL TIMER (PIT)

The 8253 PIT (U34) near the lower middle of the system board (Fig. 1-11) receives the divided 2.386 MHz clock signal and develops special time of day and date signals which are maintained as long as power is applied to the machine. These time and date signals can also be stored in mass memory with a file you have developed so you know which version of a program you are accessing when you reload the file into temporary memory on the system board. It also produces speaker pulses for generating sound.

8237 DIRECT MEMORY ACCESS (DMA) CONTROLLER

Below U34 in Fig. 1-11 is an 8237 DMA Controller (U35). This special purpose microprocessor enables large data block transfers between mass storage (disk drives or other I/O) and the internal memory on the system unit board without CPU involvement. It can handle data blocks up to 64K bytes in length.

One of the four DMA channels of U35 combines with one of the three counter/timer channels of U34 to produce a refresh signal for the temporary memory chips and any additional memory mounted on expansion boards and plugged into one of the five expansion sockets.

8255 PROGRAMMABLE PERIPHERAL INTERFACE (PPI)

The IC below U35 in Fig. 1-11 is the 8255 PPI (U36). This smart peripheral device allows external communication with the CPU. It is a parallel I/O chip with three ports that can be configured by a software command to function as either input or output. The PPI is used to read the system board configuration switches to determine how much memory is installed, the number of disk drives, and the type of display screen being used. It also accepts data input from the keyboard.

READ ONLY MEMORY (ROM)

The column of large chips beginning with the empty socket for U28 holds 40K bytes of ROM. The empty socket was originally designed to hold cassette BASIC but wasn't required in the final configuration. Four 8K ROMs contain a large cassette BASIC high level language interpreter program. A fifth 8K ROM chip below U28 contains the ROM basic input/output system (BIOS) that enables CPU communication with the system circuitry. It handles video display graphics, a time-of-day clock, printing to the screen cassette operations, and printer and asynchronous device communications. Included in this chip is a self-test program that checks the functionality of the PC during power-up. It also contains a minifloppy disk bootstrap loader to cause a floppy drive to load mass storage programs onto the temporary memory on the system board.

RANDOM ACCESS MEMORY (RAM)

The temporary memory is the RAM mounted in four columns of nine chips each at the far left top of Fig. 1-11. Eight chips in each column make up the 8-bit data word. Early versions of the PC contained four columns of 16K x 1-bit ICs with a single column of chips providing the minimum 16K system and four columns of chips providing the maximum configuration of 64K RAM. After late spring of 1983, each PC was designed to hold 64K x 1-bit RAM chips providing working memory of between 64K and 256K.

In both cases, the ninth chip in each column is a special RAM used to store the parity value for each byte of stored data. During storage, parity circuitry determines the number of logic 1s in the word and adds enough (1 or 0) to make the number of 1s an even number. Thus 10010001 in

the stored word would cause the circuitry to store a 1 in the parity RAM thus ensuring that the total number of 1s is even. When a word is read from RAM memory, parity check circuitry computes what the parity should be and compares its value (1 or 0) with the value stored in the parity RAM. If the two values are not the same, a non-maskable interrupt called parity error occurs causing the CPU to produce a display that reads PARITY CHECK and then halts system operation.

POWER SUPPLY

The switching power supply inside the system unit provides up to 64 watts of energy to operate the computer, its internal disk drives, and the adapter cards plugged into the system board expansion slots. It can also pass 120 VAC through to a socket on the rear of the system unit for powering a monochrome display.

The supply receives 120 VAC at 50/60 Hz through the line cord and generates +5, -5, +12, and -12 volts as output. It is fused and includes a power sensing device that automatically cuts off power to the PC if too much or too little voltage is detected. It also shuts down if an overvoltage or overcurrent condition occurs because of a short on the system board, on an expansion board, or in the disk drives.

Two 6-pin connectors provide power to the system board, and two connectors provide power to each internal disk drive assembly.

In the next chapter, you will read about the role of each of the previously described chips in the PC system and gain an in-depth understanding of how the IBM personal computer system operates. Chapter 2 takes you deep into the circuitry to gain a detailed understanding of the system you are about to repair and maintain.

2

Detailed System Operation

From the moment power is applied to the IBM PC, the electronics inside this machine synthesize into a functioning system whose purpose is to serve you. To a novice this seems like magic, but to the informed it's total logic–digital logic. This chapter describes the operation of the IBM in detail. Since Chapter 2 complements the SAMS COMPUTERFACT schematic diagrams, it will be helpful if you have these before you as you read the chapter. Some of the circuit schematics in the COMPUTERFACT are simplified in this text for ease in understanding a particular signal or data flow. References to COMPUTERFACT pages will be indicated as "CF page xx."

FOREWORD

The schematic symbols used in this book were adopted from those found in SAMS COMPUTERFACT CSCS2. A box enclosing a number represents a common test point. Because printing the "not" bar over signal labels is awkward in a text such as this, the active low representation of a signal will be indicated by the symbol * following the label. Thus, an active low reset signal will be represented as RESET*. The remaining symbols should be familiar to the reader. A description of the signal labels is provided in *Appendix C*.

THE POWER SUPPLY

When you reach around to the right side of the PC and rock the power switch to the ON position, energy flows from the power supply out across the system board like brightness returning to the earth as a large cloud passes by overhead.

Figure 2-1 is a block diagram of the switching power supply. Rocker switch S1 at the top left of CF page 8 allows 120 VAC across power cord P1 to be applied to the AC input board. It also provides 0.75 amp of filtered 120 VAC to a receptacle at the rear of the power supply for powering the IBM monochrome display. Up to 800 mA of current begins to flow through P2-J2 into the AC input board building a power field around AC line choke L1 (CF page 39) and passing through the primary input voltage protection fuse F1 and RF choke L3. Power is being felt at the J4 and P3-J3 connectors and the

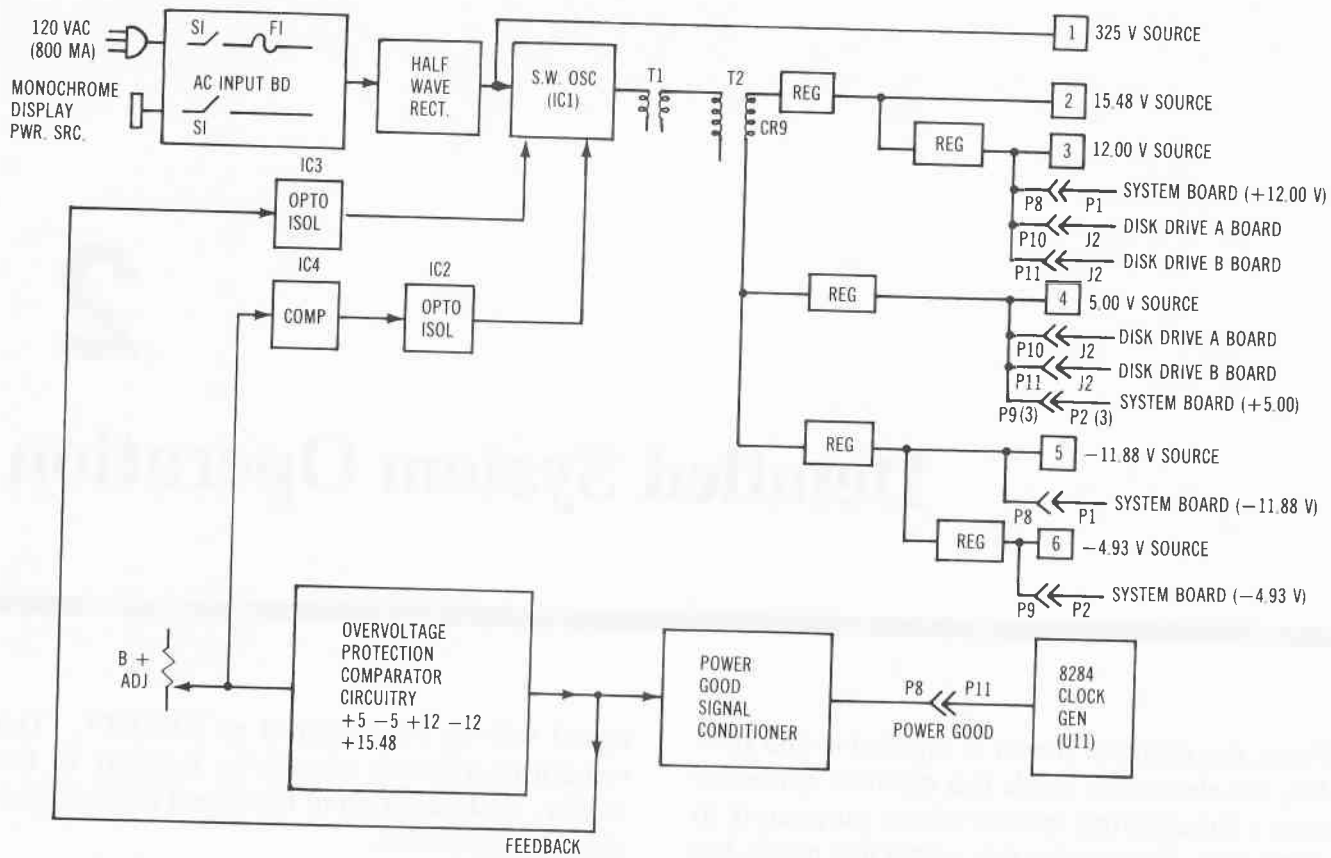


Fig. 2-1. Switching power supply block diagram.

combination of diodes CR1 and CR2 and capacitor C1. A 12 volt sawtooth waveform develops at the intersection of CR1 and C1. This signal becomes the 325V source (top right of CF page 9) that is fed forward as a reference for the oscillator circuitry on CF page 52.

At the same time, current is flowing at the box 35 junction through the ladder network R5, R6, R7, R8, R66, and R67 down CF page 8, across page 53 and up page 52 into the pulse transformer circuitry near T2. Current is also flowing horizontally from box 35 across the RC network of C2, C1, R69, R70, R71, R72, R73, R74, R75, and C10 into pin 1 of oscillator IC1.

Switching oscillator IC1 begins to conduct generating a 1 volt rectangular wave signal that is passed out pin 15 (CF page 53) to the base of oscillator driver Q2 causing Q2 to conduct producing an inverted signal on its collector. An 80 volt signal is felt across the primary of oscillator transformer T1.

The secondary of T1 is applied through the oscillator output circuitry of CR5 and Q1 to the primary of pulse transformer T2. The difference between the potential at the top and bottom of T2 primary is 143 volts as shown on CF page 52. The tapped secondary of T2 yields several voltage signals that are applied to regulator circuits to produce the switched voltages required by the system.

From the top of the T2 secondary, 37.15 volts (measured from T2 top to ground) is applied across diode CR9 to produce 15.48 volts source potential used by IC4, IC5, IC6, the B+ adjust circuitry, and cooling fan motor M1. This same 15.48 volts is also felt on the collectors of regulator Q7 and regulator driver Q6 producing up to 2.0 amps of current at 12.00 volts source potential at test point TP10 and connectors P8, P10, and P11. This 12 VDC level is used to power the system's dynamic memory and the internal 5¼ inch diskette drive motors.

From another tap, 13.00 volts AC is applied to the anode of CR10 producing a 5.10 volt signal that is felt at AC line choke L1 yielding up to 4 amps of 5.00 volts DC used throughout the PC system. This 5.00 volts is used as a reference by IC4, IC6, and power good driver Q4, and is felt at connectors P9, P10, and P11.

The bottom of T2 secondary is applied to the regulator circuitry of diode CR11, capacitor C27, and resistor R50 to produce a -11.88 volt source used as a reference by IC5 and felt at test point TP11, power-supply-to-system-board-connector P8, and on the input to regulator IC7. This -11.88 VDC is described as -12 VDC. Up to 0.25 amp of -12 VDC can be used with the +12 VDC by the system to power the EIA drivers on the communications adapters.

The -11.88 volt potential is applied to pin 2 of voltage regulator IC7 producing -4.93 volts used as a reference by IC6 and felt at test point TP9. This regulated voltage is the system -5 VDC applied through connector P9 and plug P2 to the system board provide dynamic memory bias voltage. The -5 VDC level tracks the +5 VDC and +12 VDC at power-on. It has a longer decay than the +5 VDC and +12 VDC levels at power-off.

Power supply output source voltages 5.00 volts, 12.00 volts, -4.93 volts, and -11.88 volts are monitored by the overvoltage comparator circuitry of IC5 and IC6. The 15.48 volt source is monitored via its use by the B+ adjust circuitry, and IC4, IC5, and IC6. Should an overvoltage or overload occur, the output of the overvoltage protection (OVP) circuitry of IC5 and IC6 drops to a low value reducing the LED intensity of opto isolator IC3. This shuts down switching oscillator IC1 dropping all source voltages to 0. All four of the primary voltages (+5, -5, +12, -12 VDC) are overvoltage, overcurrent, open-circuit, and short-circuit protected so catastrophic damage doesn't occur on the system or adapter boards should one of these conditions occur. An overvoltage, power-supply shutdown occurs if either the +5 or +12 VDC outputs exceed 200 percent of maximum rated voltage. The supply also shuts down if current through any output exceeds 130 percent of nominal voltage.

POWER GOOD SIGNAL

About 100 ms after the source voltages have reached their minimum sense level, the output of the OVP circuitry shown in Fig. 2-2 is felt on the base of power-good amplifier Q3 (CF page 53) causing Q3 to conduct placing 4.40 volts on the base of power good driver Q4 which also conducts. The collector output of Q4 sequences from 0.0 VDC (0.0 VDC to 0.4 VDC) capable of sourcing 500 μA to a TTL-compatible logic high (2.4 VDC to 5.5 VDC) capable of sourcing 60 μA. This nominal +5 VDC (measured 4.99) signal is called *power good*. Power good is applied through connector P8-P1 to pin 11 of clock generator (U11) on the system board initiating the awakening of the PC electronics. If the source voltages drop too low for Q4 to conduct, power good goes low disabling the clock at pin 8 of U11.

8088-BASED IBM PC SYSTEM

The IBM PC computer is constructed around the powerful 8088 central processing unit (CPU). The Intel 8088 (U3) shown in Fig. 2-3 is a third-generation microprocessor. This chip has 20 address lines so it can directly address 1 million bytes of memory. Sixteen of the address lines are used to access up to 64K of I/O memory. The I/O of the PC is memory mapped for easy access by CPU U3 and by application software.

The instruction/function format of the 8088 is identical to the 8086 microprocessor. The two machines differ in the size of the data bus. While the 8086 has a 16-bit data bus, U3 has an 8-bit external data bus. U3's standard 40-pin dual in-line package can be driven by a single +5 volt power source.

The CPU uses a time-multiplexed address and data bus format that permits several device pins to serve dual functions. During the machine cycle, eight address lines become data lines. Some of the control pins can also serve dual functions as determined by the strapping of the minimum/maximum (MN/MX) pin 33. In the IBM PC, MN/MX pin 33 is strapped to ground

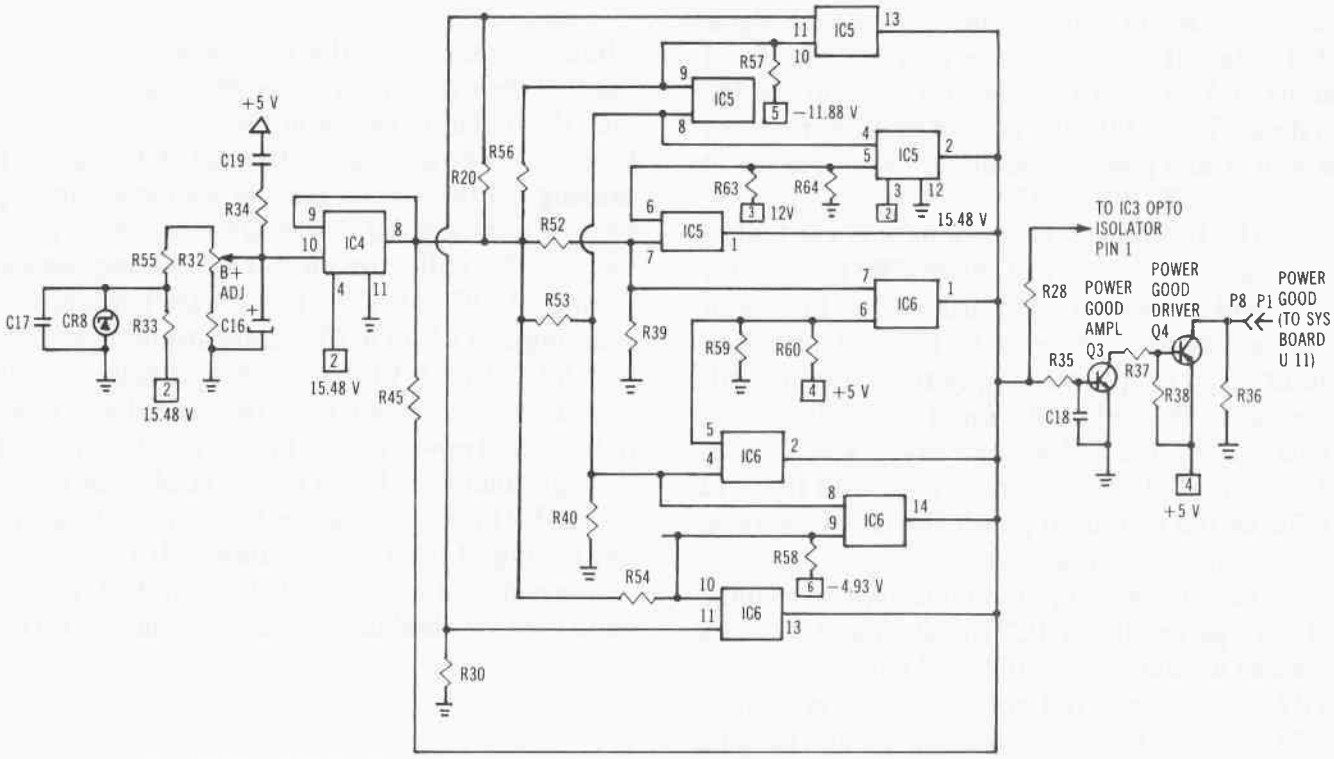


Fig. 2-2. Power good generation circuitry.

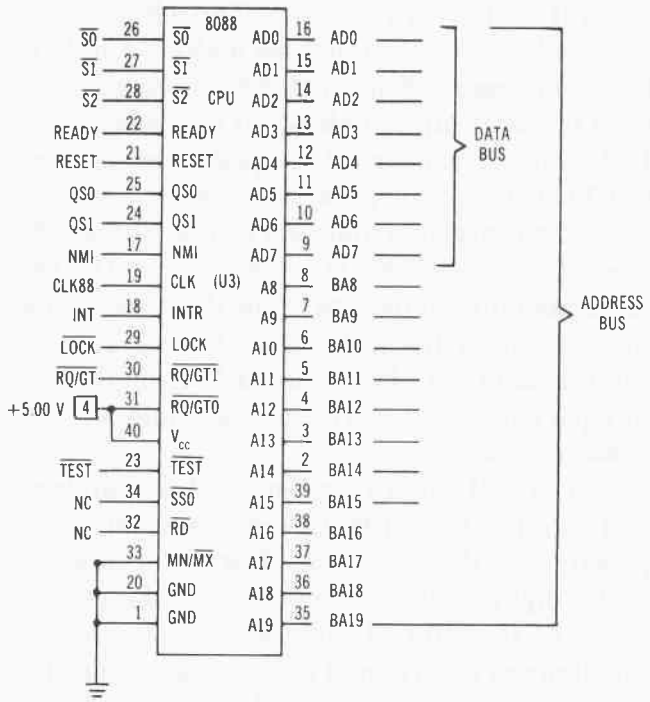


Fig. 2-3. Pin allocation for the 8088 CPU (U3).

placing U3 in maximum mode as shown in Fig. 2-4. In this configuration, U3 can support the 8087 local instruction set extension processor. An 8288 bipolar bus controller provides sophisticated bus control and command functions.

Basic to causing U3 to operate are the CLK88, READY, and RESET signals from the 8284 clock generator (U11).

Clock Generation Circuitry

The IBM PC's CPU requires a clock signal with fast rise and fall times (10 ns maximum). Its low value must be between -0.5 and +0.6 volt, and its high value must be between +3.9 volts and Vcc. The 8088 CPU incorporates dynamic cells. Therefore, a minimum frequency of 2 MHz to the CPU is required to retain the state of the machine.

With the four source voltage potentials being felt across the electronics of the system board and adapter cards, the PC begins to stir

(electronically of course). Figure 2-5 shows that a 14.31818 MHz series resonant fundamental mode crystal has been connected across pins 16 and 17 of the 8284 clock generator and driver (U11) to produce the basic operating frequency. This clock crystal oscillates at three times the CPU frequency.

Input F/C (pin 13) is strapped to ground to permit the CPU's clock to be generated by the crystal rather than by an external frequency clock reference.

Because the oscillator will fail if the attenuation of the feedback circuit reduces the loop gain to less than one, series resistors R25 and R22 are connected to the input to the internal crystal oscillator. Resistors R13 and R22 stabilize the 14.31818 MHz reference signal applied to U11's internal crystal oscillator. The 14.31818

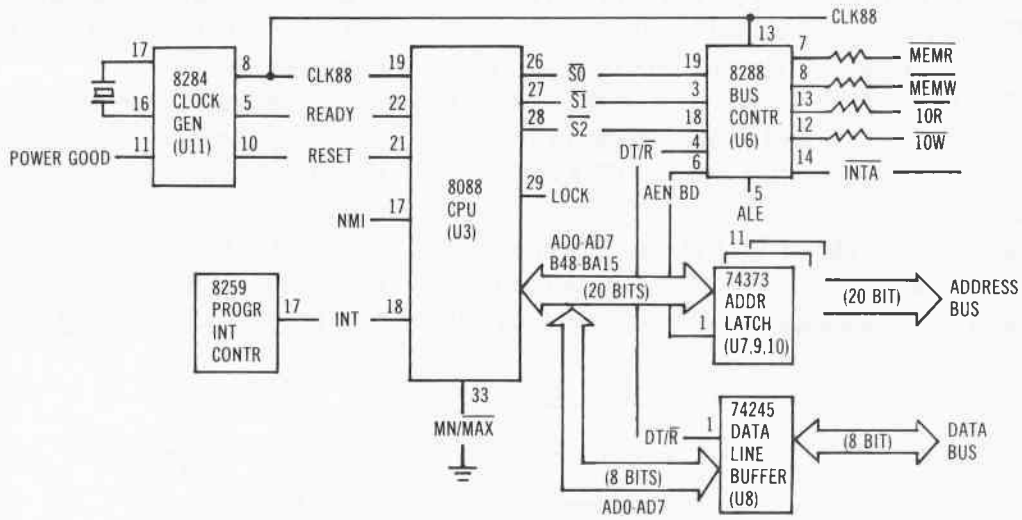


Fig. 2-4. 8088 maximum mode configuration.

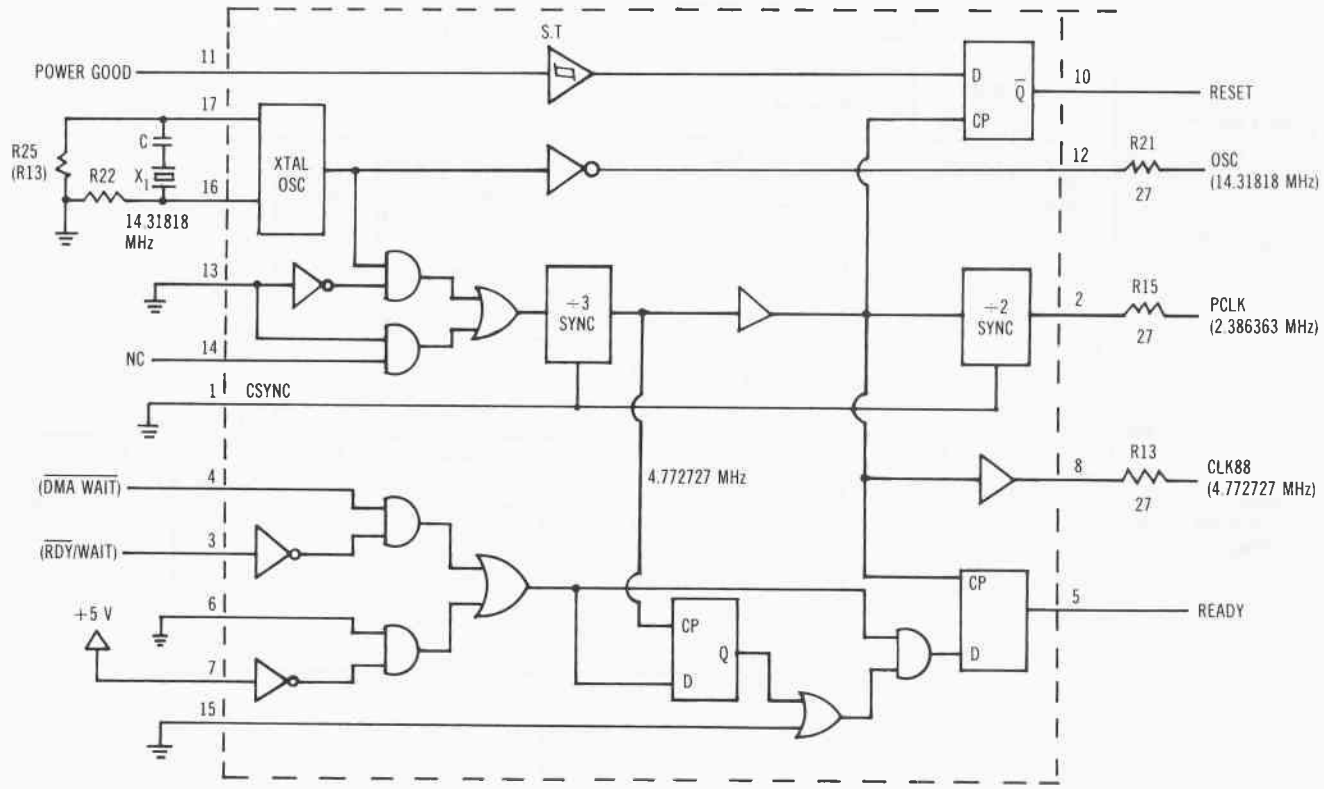


Fig. 2-5. 8284 clock generator (U11) circuitry.

MHz clock pulses coming out of the oscillator are buffered through an inverter and are made available at pin 12 as a 14.31818 MHz system oscillator signal OSC. The OSC clock signals are used by the color/graphics adapter board to develop synchronization and horizontal scan signals.

The output of the internal crystal oscillator in U11 is also applied to AND/OR and divide-by-three logic to produce a 33 percent duty cycle 4.772727 MHz clock signal optimized for the system's CPU at maximum frequency. The 4.772727 MHz clock is used to step the lower D flip-flop in Fig. 2-5 and is also passed through two buffers to become the system's CLK88 signal on output pin 8.

Since the state of U11 is indeterminate at power on, an external sync provides synchronization of the clock signals to the external crystal oscillation. Pin 1 input CSYNC is strapped low, so once the chip and crystal activate, the next positive clock from the frequency source starts the chip output clock generation.

CLK88 is applied directly to the 8088 CPU U3 (hence the label "Clock 88") and the 8087 math coprocessor (U4 on CF page 2), and 8288 bus controller (U6 on CF page 4), as shown in Fig. 2-6. It is also connected to pin 13 of 74LS244 tristate octal buffer (U15 on CF page 59) where it becomes the CLK signal for the expansion slots.

The output of the first buffer after the divide-by-three logic in Fig. 2-5 becomes a clock signal for two D flip-flops and a divide-by-two circuit. The divide-by-two circuit produces a 2.386363 MHz peripheral clock signal, PCLK, at output pin 2. PCLK has a 50 percent duty cycle and is connected to Fig. 2-6 pin 9 of 74LS175

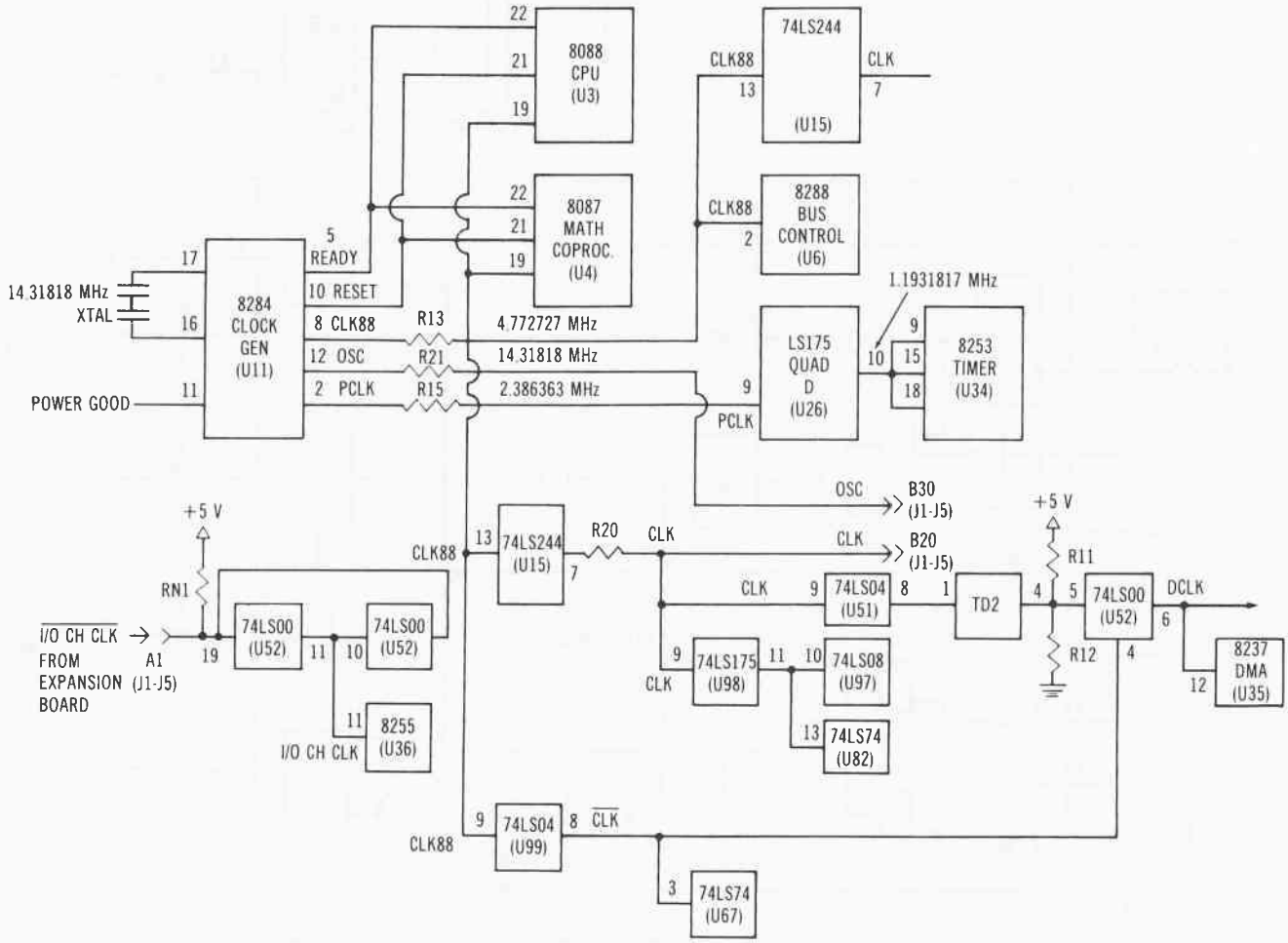


Fig. 2-6. Block diagram of clock circuitry in PC.

quad-D flip-flop (U26 on CF pages 4 and 57) which further divides it down to 1.1931817 MHz and passes it to pins 9, 15, and 18 of the 8253 programmable interval timer (U34).

All three output clock signals (OSC, PCLK, and CLK88) pass through individual 27 ohm series resistors which act to eliminate ringing.

Power good from Q4 in the switching power supply passes through Plug P1 on the system board to pin 11 of U11 (CF page 4). The power good input to U11 is not recognized until it drops at least below 2.6 volts. This input must remain below 1.05 volts for at least 50 μs after Vcc reaches the 4.5 volts minimum supply voltage–the maximum signal period is 124 μs.

Reset

As shown in Fig. 2-5, power good enters the RES* input to the 8284 clock generator and driver U11 to activate an internal Schmitt trigger producing a pulse input to the RESET D flip-flop. The flip-flop is used to synchronize the power good input with the CPU clock to generate the RESET signal to the CPU and a general reset to the entire system. Power good has no effect on the clock circuits within U11.

After the Schmitt trigger has fired, the trailing edge of the next 4.772727 MHz CLK88 pulse clocks the Schmitt trigger output into the flip-flop and shortly thereafter, the Q* output goes high sending a RESET signal out pin 10 to the components shown in Fig. 2-7.

The RESET signal enters the 8255 programmable peripheral interface (U36) on pin 35 clearing its control register and setting all ports to the input mode.

The signal also enters the 8087 math coprocessor (U4) on pin 21 clearing its internal registers and preparing it for 8088 support operations.

When RESET is felt on pin 21 of the 8088 CPU (U3), its internal registers are zeroed, and its code segment (CS) register is preset to 0FFFFH, the initialization address for future operations.

The active high RESET signal enters the 8237 DMA Controller (U35) on pin 13 causing its internal timing and control section to disable all four direct memory access channels, clearing the command status, request, temporary registers, and the first/last flip-flop. It also sets the mask register and places the device in an idle cycle mode. The 8237 programmable DMA controller (U35) will be covered shortly.

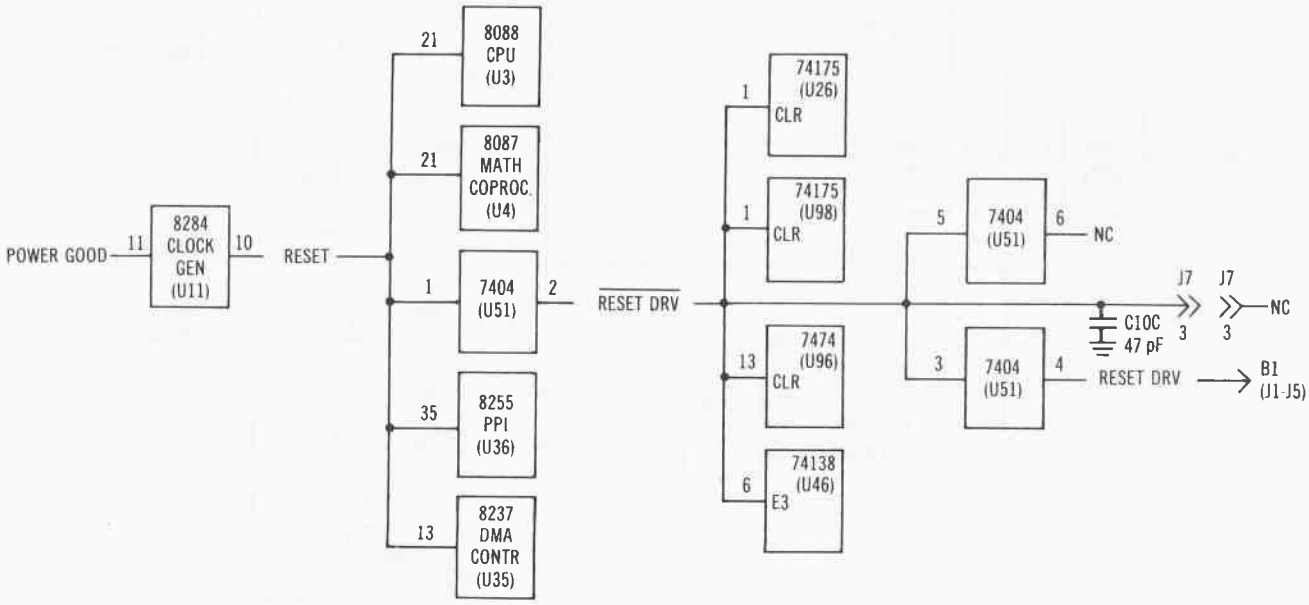


Fig. 2-7. The RESET circuitry.

RESET also passes through pin 1 of a 74LS04 hex inverter (U51) to become the active low reset drive (RESET DRV)* signal at its pin 2 output, as shown in Fig. 2-8. (RESET DRV)* is used to reset or initialize system logic upon power-up or during a low line voltage outage. It is applied to the clear input (pin 1) of two 74LS175 quad D latches (U26 and U98) and to the clear input (pin 13) of 74LS74 dual D latch U96 where it is used to reinitialize the non-maskable interrupt (NMI) circuitry. (RESET DRV)* is also applied to the active high E3 enable input (pin 6) of 74LS138 decoder (U46). (RESET DRV)* is terminated with a 47 pF capacitor (C10C) and is connected to an unused pin of keyboard connector J7.

Finally, (RESET DRV)* from U51 pin 2 is passed back into its pins 3 and 5. The pin 3 input becomes an active high RESET DRV signal pin 4 output and is made available to the expansion boards via pin B2 on J1 through J5. The pin 6 output of hex inverter U51 is not used.

8088 CPU OPERATION

With the system in a reset condition and the clock oscillator running, the machine pulses into action. The key to all the activity is the 8088 CPU (U3).

As shown in Fig. 2-9, the 8088 CPU incorporates two separate processing units: an execution unit (EU) and a bus interface unit (BIU). Standard microprocessors with a sequential CPU architecture execute programs by fetching an instruction, executing the instruction, and then fetching the next instruction in a long time-consuming sequence. The logic in the CPU must wait for the fetch and instruction decoding operations before execution can begin. The control and arithmetic logic must spend a lot of overhead time waiting. Intel eliminated this wasted time by partitioning the 8088 CPU into two independent sections.

Operating in a pipelined fashion, the BIU fetches and temporarily stores instructions in a

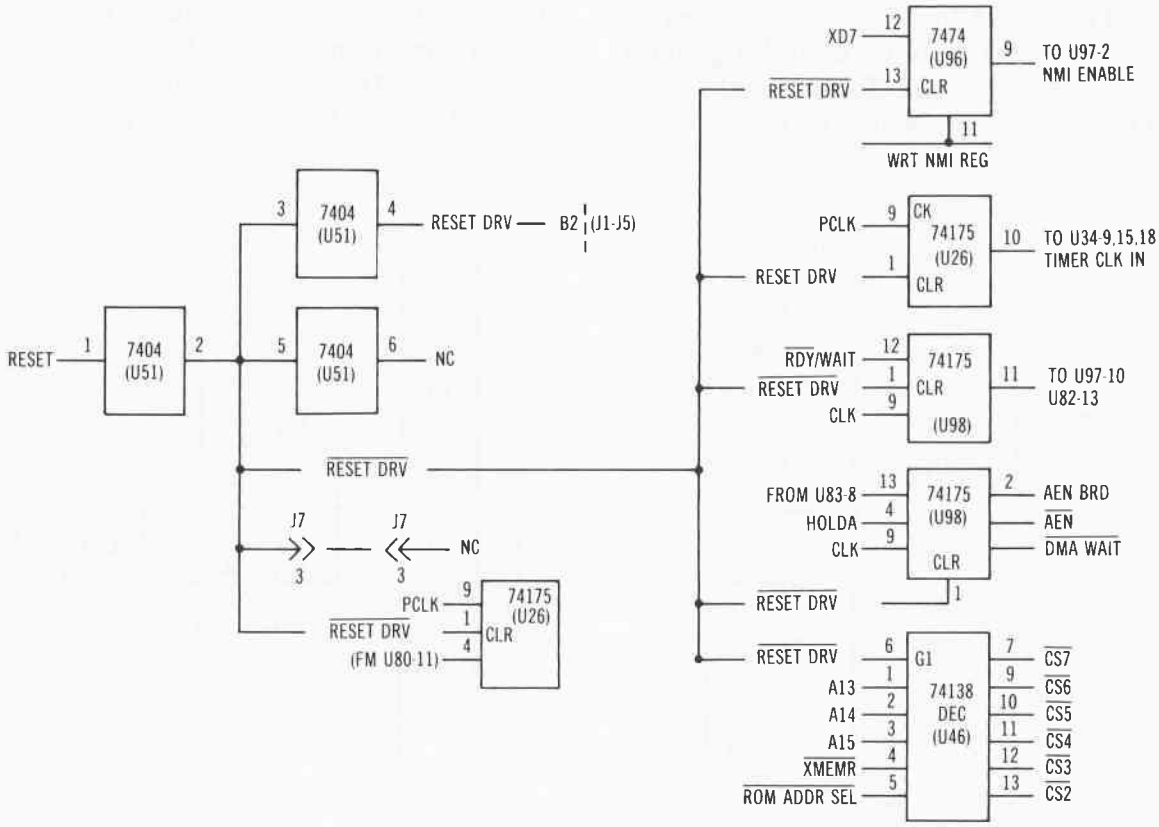


Fig. 2-8. Reset drive circuitry.

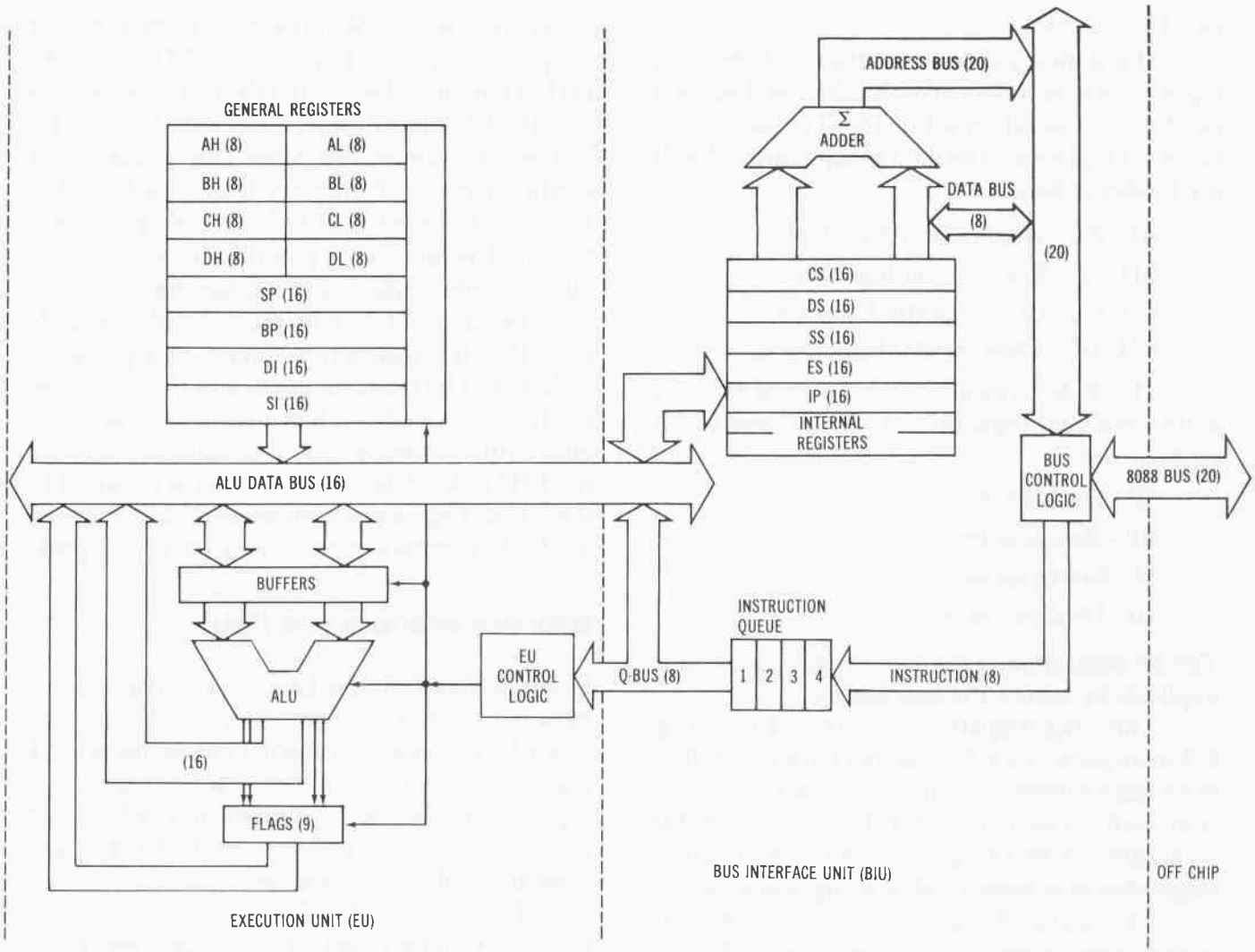


Fig. 2-9. The 8088 microprocessor block diagram.

queue of four registers. It also inputs and outputs data or operands. The EU is isolated from the chip I/O pins by the BIU. The EU executes the instructions read into a 4-byte instruction queue by the BIU. The BIU writes the results. When 1 byte of the BIU instruction queue is empty, the BIU executes an instruction fetch cycle to access one instruction object code byte per bus cycle. If the EU needs the data bus, it issues a request for access to the BIU which completes any instruction fetch cycle and then acts on the EU request.

8088 Execution Unit

The EU executes all the instructions, provides data and addresses to the BIU, and manipulates its general and flag registers. It contains a 16-bit arithmetic logic unit (ALU) that interacts with the instruction operands from the BIU queue and its general registers. The EU also maintains CPU status and control information in a flag register. All the data paths in the EU are 16-bits wide for optimum internal information transfers. The EU interfaces with the BIU via a byte-wide instruction operand O-bus between the BIU instruction queue and the EU control logic. A second interface is via the 16-bit ALU data bus.

Eight 16-bit general registers support the operation of the BIU. Collectively acting as an accumulator, these registers are divided into two sets of four registers: the high and low byte data registers (AH, AL, BH, BL, CH, CL, DH, and DL), and four P & I pointer and index registers

(SP, BP, DI, and SI).

The upper and lower sections of the data registers can be independently addressed so each can function as either a full 16-bit register or as two 8-bit registers. The data group register labels are further defined as

AH, AL - Accumulator high, low

BH, BL - Base register high, low

CH, CL - Count register high, low

DH, DL - Data register high, low

The P & I registers can be used in most arithmetic and logic operations. These 16-bit registers are

SP - Stack pointer

BP - Base pointer

SI - Source index

DI - Destination index

The SP pointer and both index registers are used implicitly by some CPU instructions.

The Flag Register below the CPU in Fig. 2-9 incorporates six 1-bit status flags that reflect conditions following an arithmetic or logic operation, and three 1-bit control flags that enable interrupts to be recognized by the ALU allow single-step operation, or alter string operations.

The status flags can cause certain CPU instructions to execute differently depending on the state of these flags. The six status flags are

OF - Overflow flag

SF - Sign flag

ZF - Zero flag

AF - Auxiliary carry flag

PF - Parity flag

CF - Carry flag

The three control flags are

TF - Trap flag

DF - Direction flag

IF - Interrupt enable flag

With these powerful registers and flags, the EU obtains instructions stored in the BIU queue and performs arithmetic, logic, move, shift, or other operations depending on the code in the instruction word. When access to memory or peripheral devices is required, the EU causes the BIU to obtain and store the data. The addresses that the EU places on its ALU data bus are 16-bits wide. Since the 8088 can access over a million bytes of memory locations (2E20 = 1.048,576), the ALU data bus 16-bit address is relocated by the BIU to give the EU access to the full megabyte memory space of the chip.

The EU fetches instruction object code from the BIU instruction queue, interprets the code in the EU control logic, and then executes the instruction. If no instruction code is in the queue (queue empty), the EU waits patiently for the BIU to load its queue. If data is required by the EU during instruction execution, it requests the BIU to access a memory location or I/O port.

8088 Bus Interface Unit (BIU)

Upon demand by the EU, data is transferred between the BIU and memory or I/O devices. The BIU executes all external bus cycles according to command from the EU. It is comprised of segment registers, a summation adder, an instruction queue, I/O bus control logic, and some internal communication registers.

The adder in the BIU combines segment and offset values to derive 20-bit addresses giving the 8088 a direct access memory space of 1 million bytes.

8088 Instruction Queue

An internal 4-byte microinstruction pipeline storage memory called the "instruction queue" enables the BIU to prefetch and temporarily hold instructions for use by the EU with minimum machine delay. While the EU is busy executing instructions, the BIU fetches more instructions from program memory keeping its pipeline queue full. Whenever the queue has an empty byte location and the EU is not requesting the external bus, the BIU fetches another instruction to keep the queue full. Maximum program execution speeds are achieved because the EU does not have to wait for each fetch operation.

The instructions fetched by the BIU are stored in sequential ascending order in memory. If the EU transfers control to another microinstruction memory location, the BIU resets the queue, passes the new instruction through its queue into the EU, and then begins refilling the queue from that portion of program memory. Fetch operations are suspended whenever the EU requests a memory or I/O access.

8088 Segment Registers

The 1 megabyte memory space of the 8088 CPU is divided into 64K byte logical segments. Four of these segments can be directly accessed by the CPU using four special 16-bit registers. These segment registers are labelled:

CS - Code segment

DS - Data segment

SS - Stack segment

ES - Extra segment

These registers hold the starting locations of four 64K logical segments. Each segment is a logical portion of contiguous memory locations and is independent and separately addressable. A unique base address is assigned to each segment to indicate its starting point in memory space. Segment base addresses begin on 16-byte memory boundaries. Therefore, as long as the boundary base line is maintained, segments can be adjacent, overlapped, or separated by unassigned memory locations.

The CS register holds the base address of the current code segment. The BIU fetches instructions from the 64K-byte portion of memory indicated by the address in the CS register. The SS register points to the top of the stack segment. Register DS points to the current data segment, and ES points to the current extra segment of data storage.

Each of these segment registers is accessible to the programmer. Using the four currently addressable segments, a software engineer can generate a program to access code and data in other segments of memory space by changing the segment registers base addresses. The current base addresses provide 256K bytes of work space–64K for microinstruction code, a 64K deep stack, and 128K of data storage.

8088 Instruction Pointer

A 16-bit program counter called the "instruction pointer" (IP) is updated by the BIU to contain the number of bytes the next instruction is offset from the base address in the current code segment register. Thus, the IP points to the next instruction to be fetched by the BIU. When the IP value is saved on the stack in the SS register, it is first adjusted to point to the next instruction to be executed. Instructions are densely packed into memory. They vary in length from 1 to 6 bytes in length. As the BIU fetches an instruction, the code in the first byte (called an "opcode") identifies the type of instruction and whether there are more bytes comprising the total machine instruction.

PHYSICAL ADDRESS GENERATION

As shown in the 8088 pin allocation diagram of Fig. 2-3, the 20-bit 8088 address bus exits 8088 CPU U3 on pins 2 through 16 and pins 35 through 39. The lower 8 bits of the 8088 output (pins 9 through 16) are bidirectional. During the later half of a machine cycle these pins are bidirectional and contain data. During the address generation part of the cycle, these same pins become part of a 20-bit address bus (pins 2 through 16 and pin 39) over which the CPU communicates with the rest of the IBM PC circuitry. This time multiplexing of bidirectional lines enables larger memory address space using fewer I/O pins.

A physical 20-bit address uniquely identifies each byte in a 1 megabyte memory space accessible by the CPU. With 20 bits in each address word, a physical memory range between 00000H and FFFFFH can be directly accessed in the IBM PC.

The powerful programming capability of the 8088 enables logical addressing using the segment registers and the instruction pointer. Each logical address is comprised of a segment value indicating the first byte in that 64K segment and an offset value defining the distance in bytes from the base address. The 16-bit segment base line address is shifted left 4 bits and combined with a value stored in an offset register to form a 20-bit physical address as shown in Fig. 2-10.

Many different logical addresses can map to the same physical address. In Fig. 2-11, physical address 03B4H can be reached via a segment base address of 03B0H with an offset of 4H and a segment base address of 03ADH with an offset value of 7H.

BIU program memory accesses are achieved by generating a physical address from logical segment and offset values. The segment base address is shifted 4 bits in the BIU summation adder and is then added to the offset value stored in the IP, SP, SI, or DI registers. It can also be an effective address as shown in Table 2-1. The logical address is obtained from different sources depending on the type of memory reference. Instructions are fetched from the current segment base line by the CS register. The IP register contains the offset of the target instruction from the segment base address.

For stack operations, the SS register contains the starting address of the current stack segment. The stack pointer (SP) register in the EU holds the offset value from the top of the stack to the stack segment base address.

Memory operands are assumed to be within the current data segment, but the BIU can access these variables in other currently addressable segments. To reach these, the offset value is calculated by the EU based on the addressing mode indicated by the microinstruction. In this case, the offset is called the operand's "effective address" (EA).

Other memory references are possible and other EU and BIU registers can be caused to participate in the generation of a logical address for mapping into a physical address by the BIU summation adder.

Table 2-1. Sources for Logical Address in the 8088 CPU

Memory Reference Type	Default Segment Base Register	Alternate Segment Base Register	Source for Offset
Instruction fetch operation	CS	none	IP
Stack operation	SS	none	SP
String source	DS	CS, ES, SS	SI
String destination	ES	none	DI
Variable (except following)	DS	CS, ES, SS	effective address
BP used as base register	SS	CS, DS, ES	effective address

THE CPU BUS CYCLE

To understand the multiplexing operation, you must first recognize that the fetch, decode, execute, and data transfer actions occurring in the CPU constitute a machine, or bus cycle. The

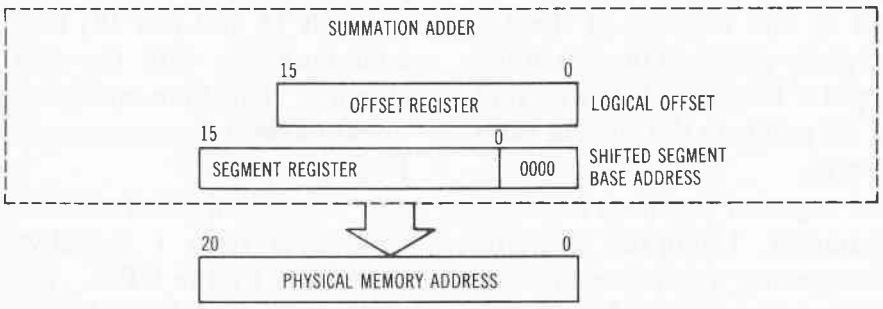


Fig. 2-10. A logical 16-bit address added to an offset 16-bit segment base address produce a 20-bit physical address.

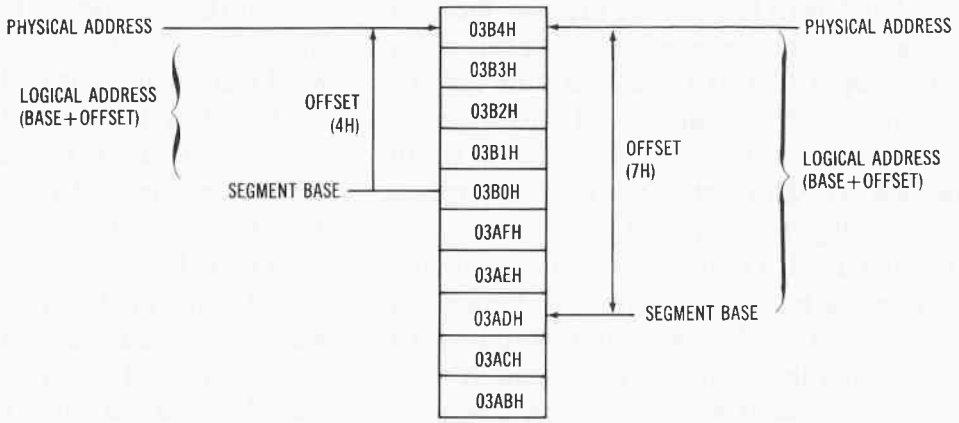


Fig. 2-11. Different logical addresses can map to the same physical address.

minimum bus cycle consists of four 8088 clock cycles. Each cycle of the CLK88 signal is called a "tick" and is represented by the letter "T." Up to six T states can be required for a single bus cycle. If the memory or I/O circuitry cannot transfer data fast enough, the 8088 CPU (U3) introduces special wait states (called TW) to cause the CPU to cycle in place by inserting additional clock ticks into the bus cycle until the selected device (memory or I/O) communicates to U3 that it is ready to continue. Fig. 2-12 shows the bus cycle associated with U3. Notice that the T-state begins on the falling edge of CLK88.

Because the IBM PC uses the 8088 in its maximum mode, any change in the status signals S0*, S1*, and S2* on pins 26, 27, and 28 of U3 during T4 indicates the beginning of a new bus cycle. These status signals start to change during T4 and become valid during T1 and T2. When T3 begins (or during TW if READY is high), these lines return to a passive state and indicate the end of a bus cycle. Status signals S0*, S1*, and S2* are used by the 8288 bus controller (U6) to generate all memory and I/O access control signals.

Once the S0*-S1*-S2* T4 change has been recognized by U3, T1 begins with the falling edge of the next clock cycle. During T1, U3 places an address code on the 20-bit time-multiplexed

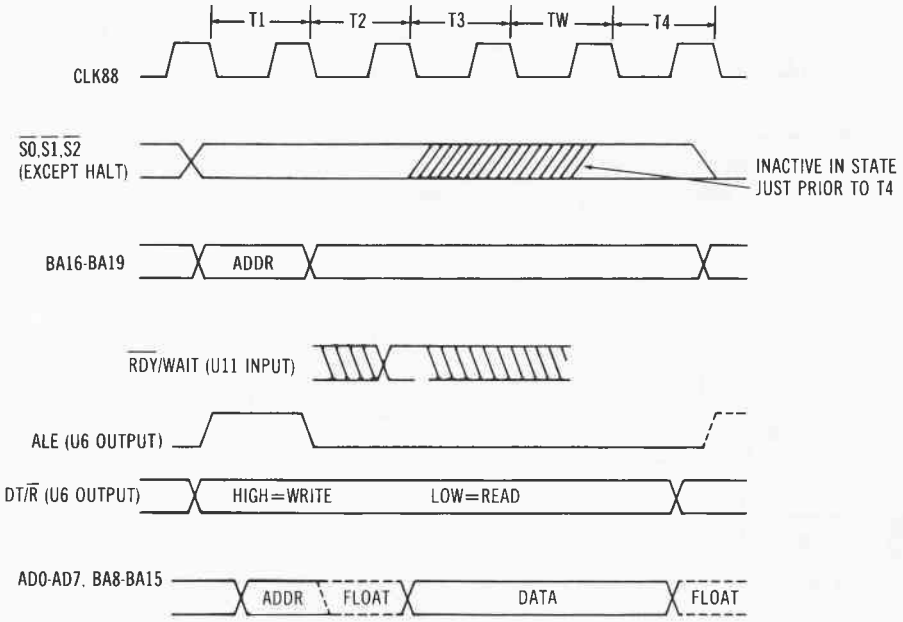


Fig. 2-12. 8088 CPU bus cycle.

address/data bus to identify the memory location or I/O device to be accessed. During this clock cycle, the top 4 bits of the 20-bit address are present on pins 35 through 38 and constitute the most significant part of the address bus (BA16 through BA19). The rest of the address appears on pins 8 through 16 and pin 39.

During T2, U3 issues an activity command on the control bus. The address information which was valid on the lower 16 bits during T1 shifts to a high impedance tristate condition if the control command is to read, or they become valid data for memory or I/O if the command is to write. Data read and write, and interrupt acknowledge control commands are active during T2. Also during T2, the upper four address lines (A16 through A19) contain bus cycle status information.

Read or write commands become valid during T3 guaranteeing sufficient time for data information to settle on the bus before command activation. Status information is present on the top 4 bits of the address bus, and data is valid on the lower 16 address/data bits.

If the memory or I/O device is slower than the 8088 CPU, it causes the READY signal from 8284 clock generator (U11 to U3 input pin 22) to be held low forcing U3 to generate TW wait states until READY returns high. The circuitry shown in Fig. 2-13 is used to generate the READY signal that passes out pin 5 of U11 into 8088 CPU U3. CF pages 2, 4, and 57 apply.

During the last TW state (or during T3 if no wait states are requested) U3 latches data on the bus. When T4 occurs, the command lines are disabled and the bus is released by any external device holding control. From a device view external to U3, the bus cycle is an asynchronous event comprised of an address to select the device followed by a read command or data and a write command. Data is copied onto the data bus by the device during a read cycle and accepted by the device from the data bus during a write cycle. The devices only control over the bus cycle is via TW wait states. The *Intel iAPX 86/88, 186/188 User's Manual* contains additional information on the 8088 bus cycle.

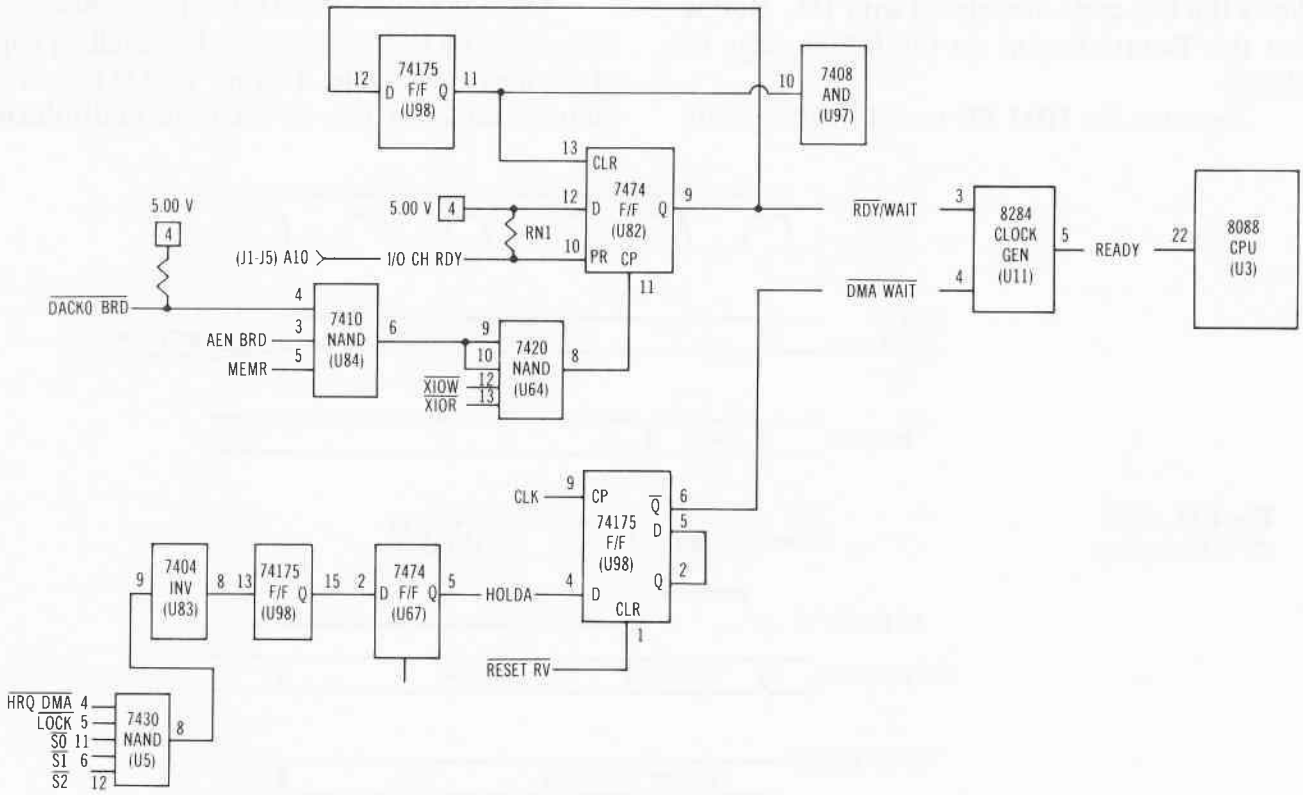


Fig. 2-13. READY signal circuitry.

Critical to the smooth operation of the bus cycle is the 8288 bus controller (U6), shown in Fig. 2-14. The 8288 is a 20-pin bipolar part that provides command and control timing signals and bus drive enable signals to the PC circuitry.

The chip has seven input pins. Pin 1, input/output bus (IOB) mode, is strapped low placing the chip in the system bus mode. In this configuration no command is issued until 155 ns after the AEN BRD line (pin 6) goes low. In this mode both I/O and memory can be shared by the 8088 CPU (U3) and the 8087 coprocessor (U4).

The 8088 status inputs (S0*, S1*, and S2*) on pins 3, 18, and 19 are decoded inside U6 to generate timed command and control signals. Table 2-2 is a decoding of the status bit inputs to U6.

The IBM PC system clock (CLK88) from the 8284 clock generator (U11) enters the 8288 on pin 2. This clock signal is used to establish when the timed command and control signals occur. Figure 2-15 is a block diagram of the bus controller circuitry.

Another input, active low address enable (AEN*) from pin 3 of 74LS175 quad-D flip-flop U98 (Fig. 2-15 and CF Page 4) enters U6 on pin 6. AEN* enables command outputs at least 115 ns after it goes active (low). When this occurs, the command outputs immediately shift to a high impedance state. Figure 2-15 shows that AEN* also connects to pin 6 (G1 enable) of 74LS138 decoder U66 (CF page 3).

Table 2-2. 8288 Status Bit Decoding

Status Bits S2*	S1*	S0*	8288 Command Generated	CPU Cycle
0	0	0	INTA/	Interrupt acknowledge
0	0	1	IOR/	Read I/O port
0	1	0	IOW/	Write I/O port
0	1	1	-	Halt
1	0	0	MEMR/	Instruction fetch
1	0	1	MEMR/	Read memory
1	1	0	MEMW/	Write memory
1	1	1	-	Passive

The other important input to the 8288 bus controller U6 is an active high address enable board (AEN BRD) control signal (U6 pin 15) from the Q output of U98 (pin 2). AEN BRD is used to enable the outputs of U6. This signal is also applied to one input (pin 15) of 74LS244 tristate buffer U15 to generate an active high AEN on output pin 5. AEN is passed to the A11 pins of socket connectors J1 through J5 (CF page 56) for use by peripheral boards. AEN BRD

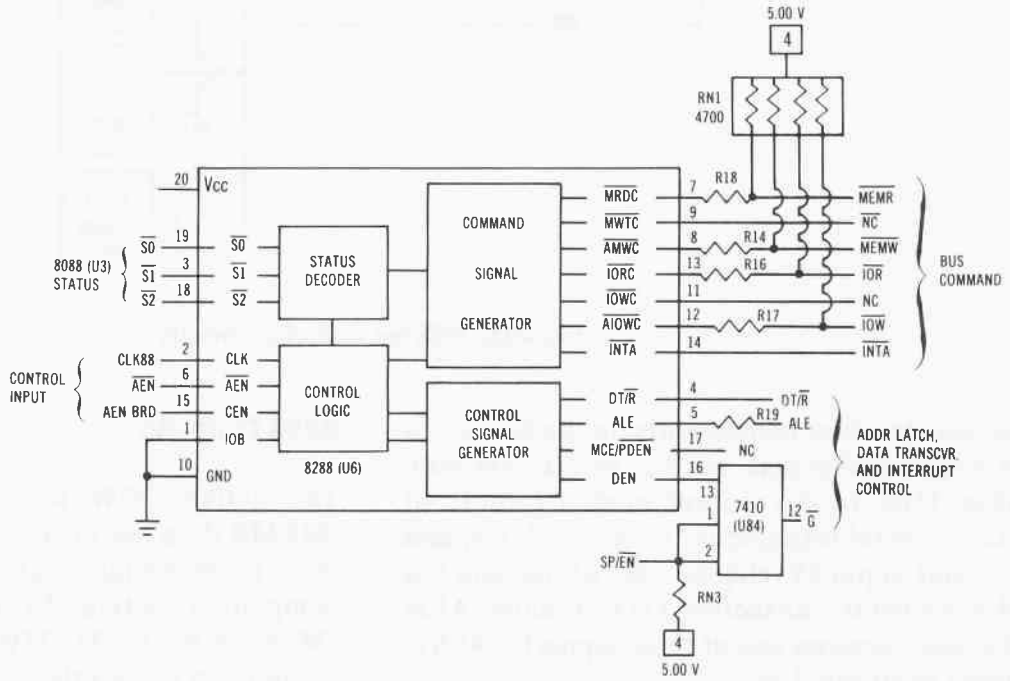


Fig. 2-14. 8288 bus controller (U6).

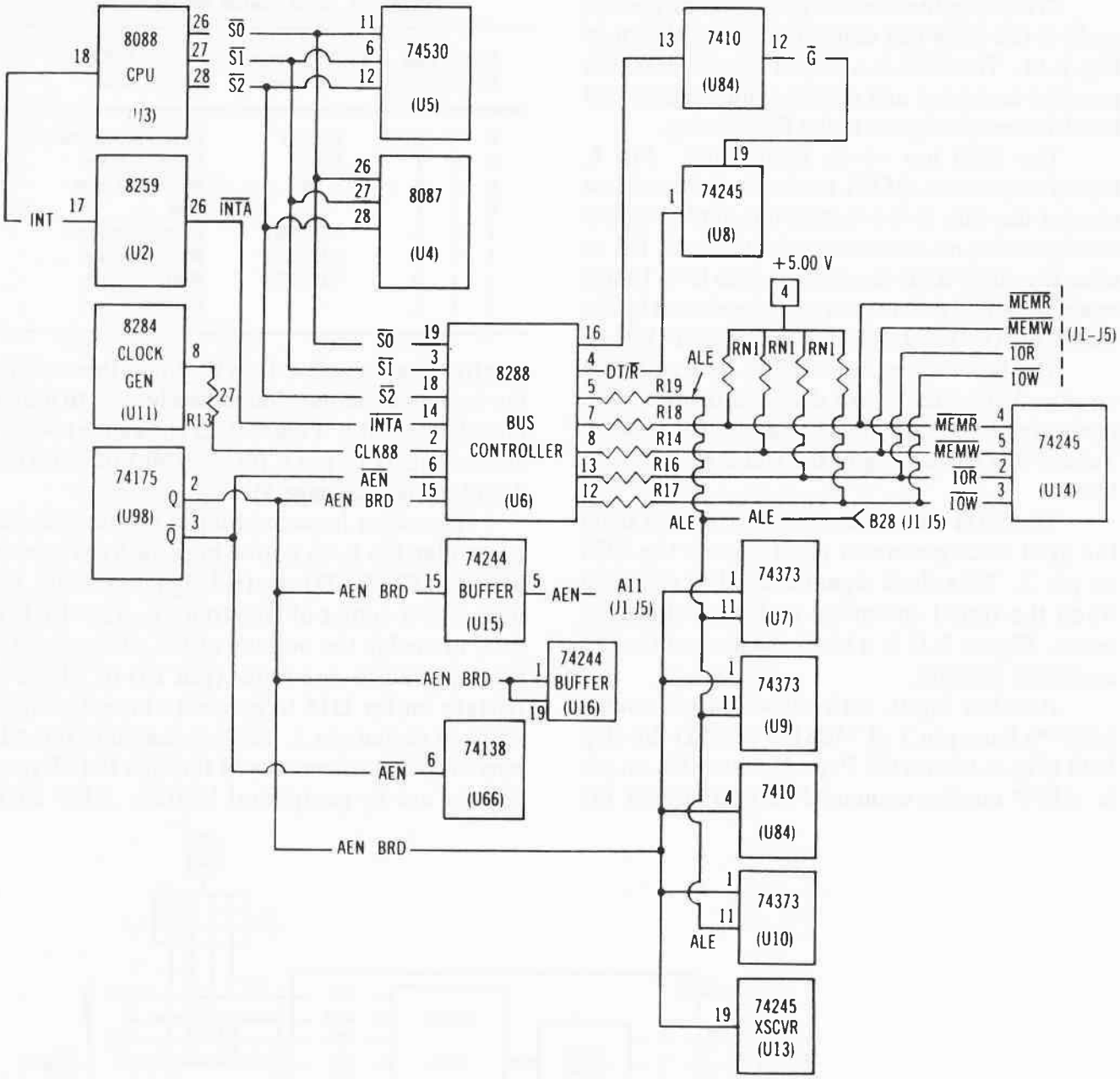


Fig. 2-15. 8288 bus controller circuitry.

connects to other components on the board: to the output enable pins 1 and 19 of 74LS244 octal buffer U16, to the output enable (pin 1) of 74LS373 octal transparent latches U7, U9, and U10, and to pin 19 (chip enable) of the data bus 74LS245 octal transceiver U13. Finally, AEN BRD also becomes one of three inputs to 74LS10 3-Input NAND gate U84.

8288 Outputs

The active LOW Memory Read command (MEMR)* on pin 7 causes the system memory to drive its stored information on the data bus. Its companion active LOW output command (Memory Write) MEMW* on pin 8 causes the memory to record the information now on the

data bus. Peripheral devices are controlled using the two active low IOR* and IOW* commands out pins 13 and 12 respectively. These command signals instruct I/O devices to drive data onto or read data from the PC data bus.

Pin 14 is the Interrupt Acknowledge INTA* active LOW output that is connected to input pin 26 of 8259 PIT U2. This signal is used to tell U2 that its interrupt has been recognized by CPU U3 and when U2 should place its vector data onto the data bus.

The outputs on the lower right of Fig. 2-15 are the control signals that interface with the address latches, data transceivers, and interrupt control. Data transmit/receive DT/R* out pin 4 of U6 connects to pin 1 of 74LS245 transceiver U8 where it determines the direction of data bus signal flow through U8. When DT/R* is high, data is sent out (transmit) 8088 U3 to I/O or memory. A low on the DT/R* pin 1 input to U8 causes data to be read from (receive) I/O or memory into U3.

The control signal out pin 5 of 8288 U6 is address latch enable (ALE). This signal is used to strobe an address into the 74LS373 address latches U7, U9, and U10 (shown on the far right of CF page 2 and in the logic diagram for the 8288 bus controller circuitry, Fig. 2-15). This ALE signal is active high and enters all three 73LS373 components on pin 11 (enable). Latching occurs on the falling (high to low) transition of the signal. ALE is also passed from pin 5 of 8288 U6 out to pin B28 of J1 through J5 interface connectors (CF page 56).

The active high data enable output from U6 pin 16 connects to the pin 13 input to 74LS10 three-input NAND U84 (CF page 4) where it is matched with the slave program/enable buffer (SP/EN)* output from pin 16 of 8259 programmable interrupt controller U2 to produce active enable signal G*. Whenever DEN on pin 16 of U6 and SP/EN* on pin 16 of U2 are high, G* is generated and becomes an active low enable to pin 19 (enable input) of 74LS245 address/data bus transceiver U8. Signal G* combines with active low DT/R* to enable signal flow from left to right through U8 as the 8088 CPU U3 or 8087 coprocessor U4 reads data from I/O or memory.

Figure 2-16 shows the timing relationships between the input and output signals of the 8288 bus controller.

The 8288 memory and I/O command signals MEMR*, MEMW*, IOR*, and IOW*, are important in proper functioning of the PC. These signals will be described in detail later in this chapter.

8253 PROGRAMMABLE INTERVAL TIMER

As shown in Fig. 2-17 (see also CF page 57), the D8253C-5 programmable interval timer U34 does three important functions in the IBM PC: it generates a time of day clock tick; it tells the DMA controller when to refresh the dynamic RAM in the system; and it helps produce sound from the machine's speaker.

The nMOS 8253 peripheral device is organized as three independent 16-bit down-counters, each with a count rate up to 2 MHz. The input clock to U34 comes from PCLK in 74LS175 quad D flip-flop U26. This causes U34 to count at a 1.1931817 MHz rate. The three output signals OUT0 (IRQ0), OUT1 (to U67), and OUT2 (TIMER/CNTR2 to U63) occur at a frequency determined by PCLK divided by a software programmable 16-bit number. U34 is used to generate accurate time delays under software control minimizing program overhead.

Upon initialization, each timer in U34 is preconfigured via a unique control word and a count quantity value sent by the 8088 CPU. These data enter U34 on pins 1 through 8 (XD0 through XD7). The initialization software writes out to U34 a *mode* control word and the programmed number of count register bytes desired.

Comprised of 8 bits, the "mode control byte" initializes a particular counter with the desired *mode*. Six modes are available:

Mode 0–Interrupt on terminal count.

Mode 1–Programmable one-shot.

Mode 2–Divide-by-N rate generator.

Mode 3–Divide-by-N square-wave generator.

Mode 4–Software triggered strobe.

Mode 5–Hardware triggered strobe.

The 8 bits of the mode control byte are allocated as follows:

Bit 0–ount in binary if equal to 0, otherwise count in BCD.

Bit 1-3–Defines which mode is desired.

Bit 4,5–Read/load sequence.

Bit 6,7–Defines which counter is to be affected.

Once the mode control word has been received, a count register in each counter is preloaded in the sequence defined by the mode control word. Each counter clocks down to zero so the value loaded into the count register decrements with each input clock pulse. Loading all zeroes into a count register causes that counter to count down from a maximum value (65,536 for binary, 10,000 for BCD). Once programmed, the counting operation is completely independent. Upon count start, the 8253 causes each counter to clock down until the preset delay value reaches zero. At this point, the zeroed counter generates a count-complete output showing it has completed its tasks.

The IBM PC software configures U34 so OUT0 operates in Mode 3 so it produces a symmetric output with equal high and low parts of each cycle. This signal becomes IRQ0, the time of day interrupt. Here, the mode control byte was 00110110 (36H) defining binary count, Mode 3, read/load the low byte, then the high byte, and identifying the counter as counter 0. Inside counter 0, its count register is loaded with the binary value 00000000 (decimal 00, hex 00H). This preset causes a maximum count delay

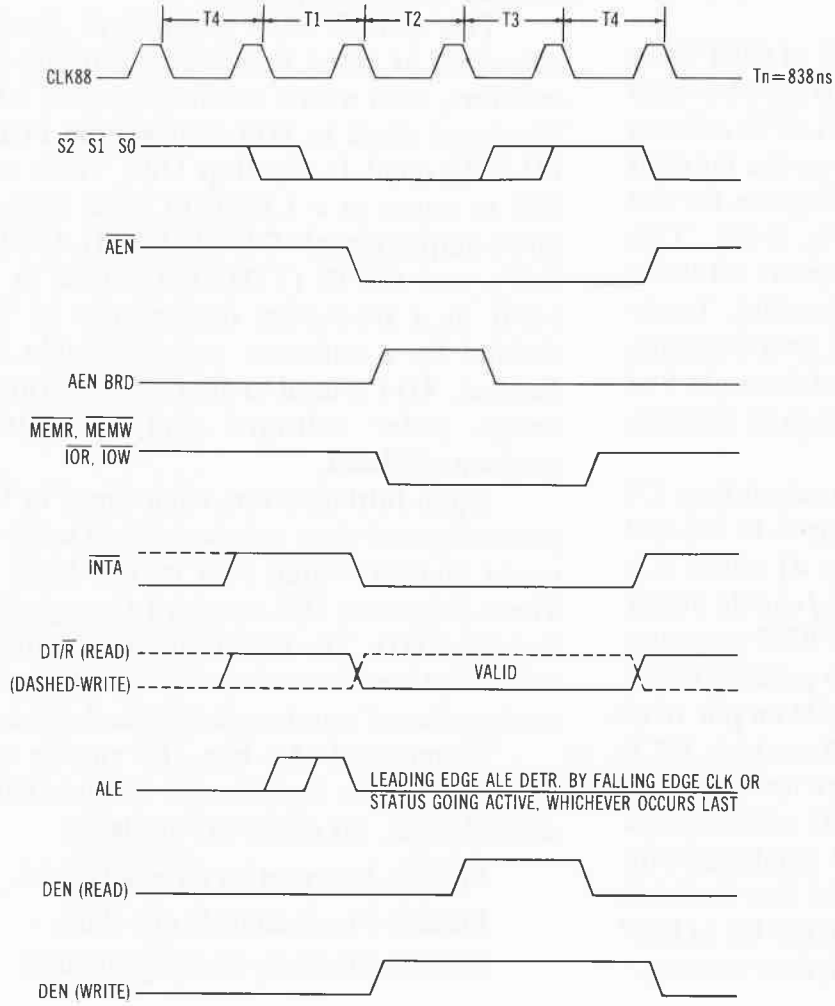


Fig. 2-16. 8288 timing waveforms.

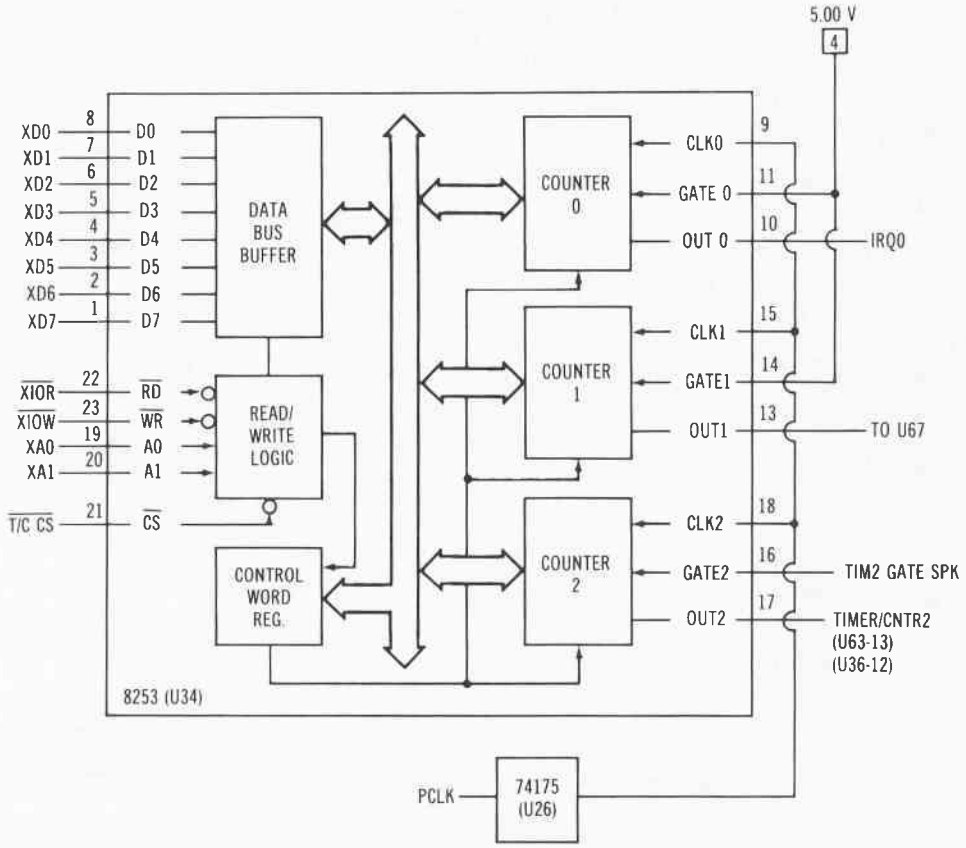


Fig. 2-17. 8253 programmable interval timer.

producing an OUT0 (IRQ0) signal approximately 18.2 times each second (1.1931817 MHz divided by 65,536 = 18.207 cycles per second).

Counter 1 was preset by loading the hex value 54H into the control word register. In binary, this is 01010100 (decimal 84) representing binary counting, Mode 2 (rate generator), read/load a low byte only, and its destination is counter 01. The counter 1 count register is preset with the binary value 00010010 (12H, decimal 18) causing this counter to produce a pin 13 OUT1 signal 66,287 times a second (1.1931817 MHz = 1193181.7 Hz, so 1,193,181.7/18 = 66287.87 cycles per second). The 66.287 kHz OUT1 signal (15 microseconds) enters pin 11 of 7474 D latch U67 (see Fig. 2-18) where it becomes DRQ0 for the 8237 DMA controller U35. DRQ0 is used to tell U35 when to refresh the dynamic RAM every 15 microseconds.

The input to the U34 word control register for counter 2 is the mode control byte 10110110 defining binary count, Mode 3, read/load low byte, then high byte, and set destination as counter 2. The preset value for the counter 2 count register is 0000 0101 0011 0011 (533 hex, 1331 in decimal). This produces an 896 Hz square wave output (1,193,181.7/1331 = 896.455 cycles per second) from pin 17 OUT2 (Fig. 2-18). OUT2 from U34 pin 17 is the TIMER/CNTR2 signal sent back to input pin 12 of the 8255 programmable peripheral interface U36 (Fig. 2-18) and on into pin 13 of 7438 two-input NAND U63. In U63, TIMER/CNTR2 is NANDed with SPKR DATA from pin 19 of 8255 PPI U36 to produce speaker activation signals. To make the speaker beep using the TIMER/CNTR2 896 kHz square wave, pins 18 and 19 of 8255 U36 (TIM2 GATE SPK and SPKR DATA) must be set high for the duration of the sound desired. OUT2 can be varied by presetting the counter 2 count register with different values producing different output frequencies under software control.

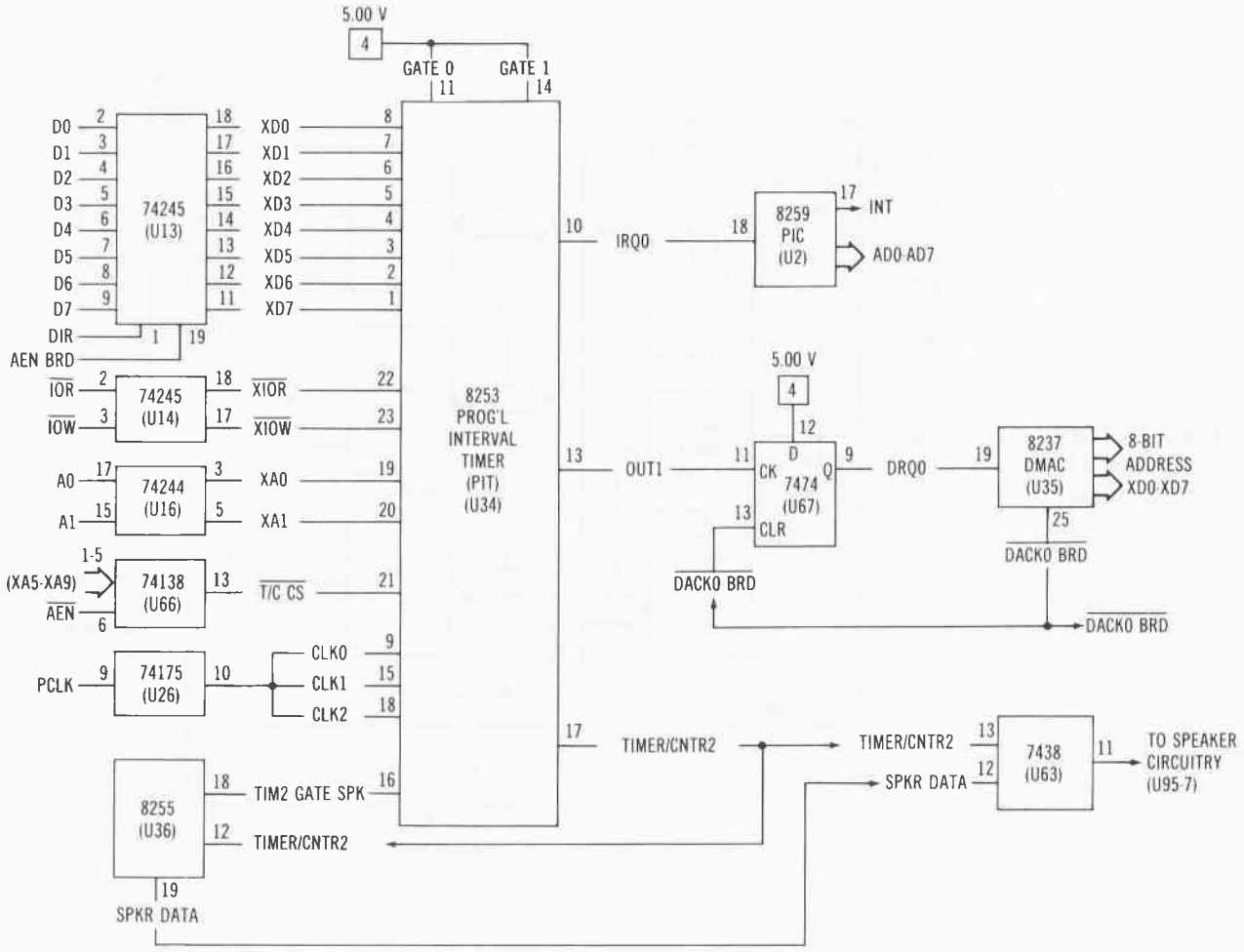


Fig. 2-18. 8253 timer circuitry.

8237 PROGRAMMABLE DMA CONTROLLER (DMAC)

The 8237 multimode direct memory access (DMA) controller (DMAC) (U35) in the IBM PC system is designed to improve system performance by allowing external devices to directly transfer information from peripheral devices (I/O) to memory or between memory and memory in up to 64K-byte blocks. The DMAC is used to suspend normal 8088 CPU operation so it can take over the system address, control, and data buses and conduct a data transfer at rates up to 1.5 megabytes per second. Innovative circuit design ensures that U35 takes bus control only at the end of a machine cycle and when LOCK* is high (not active). When DMA is active, U35 causes the address, control, and data buffers for CPU U3 to go tristate. U35 then causes the 8284 clock generator (U11) to pull READY low forcing U3 to insert wait states in its machine cycle while U35 conducts the DMA transfer.

DMAC U35 contains four independent request/acknowledge channels which can automatically reinitialize to a preprogrammed condition following an end of process (EOP*) signal. Each channel has a 64K address and word count capability. Fig. 2-19 is a block diagram overview of the 8237 DMAC. Inside the block are the vendor labelled pin descriptions. On the outside of the block are the signal labels generated by IBM. Arrows show direction of signal flow.

There are three basic blocks of control logic in Fig. 2-19. The timing and control block

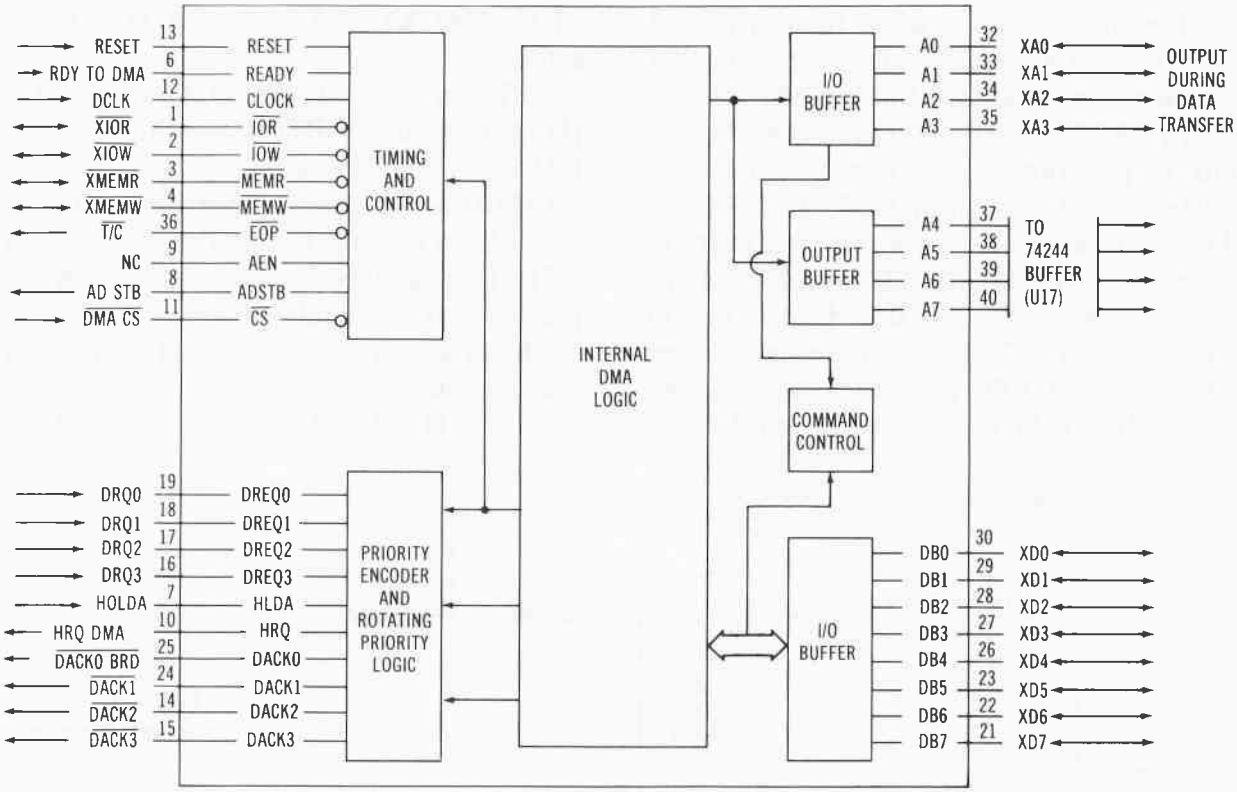


Fig. 2-19. 8237 programmable DMA controller (U35).

generates internal timing and external control signals for the chip. The internal DMA logic block decodes the various commands sent to U35 by 8088 CPU (U3) before servicing a DMA request. This block also decodes a mode control word used to select the type of DMA to be performed. The priority encoder and rotating priority logic block in the lower left of Fig. 2-19 resolves priority contention between DMA channels requesting simultaneous service.

Internal timing in the timing and control block is achieved using the 4.77 MHz DCLK (U35 pin 12) from 74LS00 U52 pin 6 in the PC clock generation circuitry.

The RESET signal on input pin 13 comes from pin 10 of 8284 clock generator U11. Activation of the power good signal from the switching power supply (described earlier) causes U11 to generate the active high RESET signal. When pin 13 goes high, it clears special registers inside the 8237 and a first/last flip-flop. Then the chip goes into an idle state until a valid DMA request arrives.

In the idle state, U35 samples the four DRQx lines (DRQ0, DRQ1, DRQ2, and DRQ3) with each DCLK input pulse to determine if any channel is requesting a DMA transfer. U35 also samples the DMA chip select input (DMA CS)* on pin 11 if the 8088 CPU is attempting to read or write to its internal registers. When (DMA CS)* and HOLDA (pin 7) are low, U35 enters a program state in which the CPU U3 can access the internal registers of U35. Address lines XA0 through XA3 act as inputs to U35 to select which internal registers will be written into or read from. The XIOR* and XIOW* signals on pins 1 and 2 are used to select and time the read or write operations.

In the program state, special software commands can be executed by U35 using sets of addresses and the (DMA CS)* and XIOW* signals. The software commands include clear first/last flip-flop and master clear.

As shown in Fig. 2-20, four DMA request channels connect to U35 (pins 16, 17, 18, and 19). These request lines are individual asynchronous

channel request inputs used by peripheral circuitry to obtain DMA service. They are in a fixed priority with DRQ0 holding the highest priority (DRQ3 lowest). The polarity of these signals is programmable. These signals are initialized to active high by RESET. DRQ0 is used to produce a memory refresh cycle every 15 microseconds. DRQ1, DRQ2, and DRQ3 are available on the expansion slots for use in I/O data transfers. DRQ2 is used to initiate floppy disk transfers. DRQx must be held until the corresponding DACKx acknowledge (DACK0 BRD, DACK1, DACK2, and DACK3) goes active.

When one of the DMA request lines (DRQ0 through DRQ3) goes high requesting DMA service, U35 outputs a hold request DMA (HRQ DMA) signal out pin 10 to pin 1 of the 74LS04 hex inverter U99 in the upper left of Fig. 2-21. This signal becomes an active high input to pin 1 of 74LS74 dual-D latch U67 clearing the latch for activation by a signal on pin 2 a few microseconds later.

The HRQ DMA output on pin 3 of U52

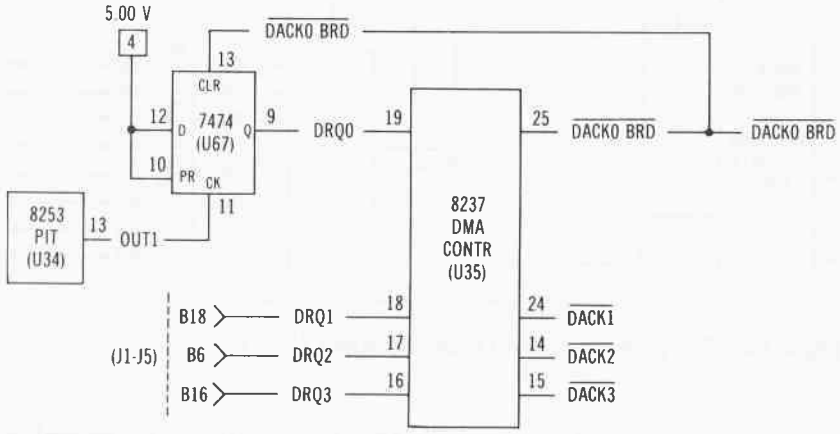


Fig. 2-20. DMA request circuitry.

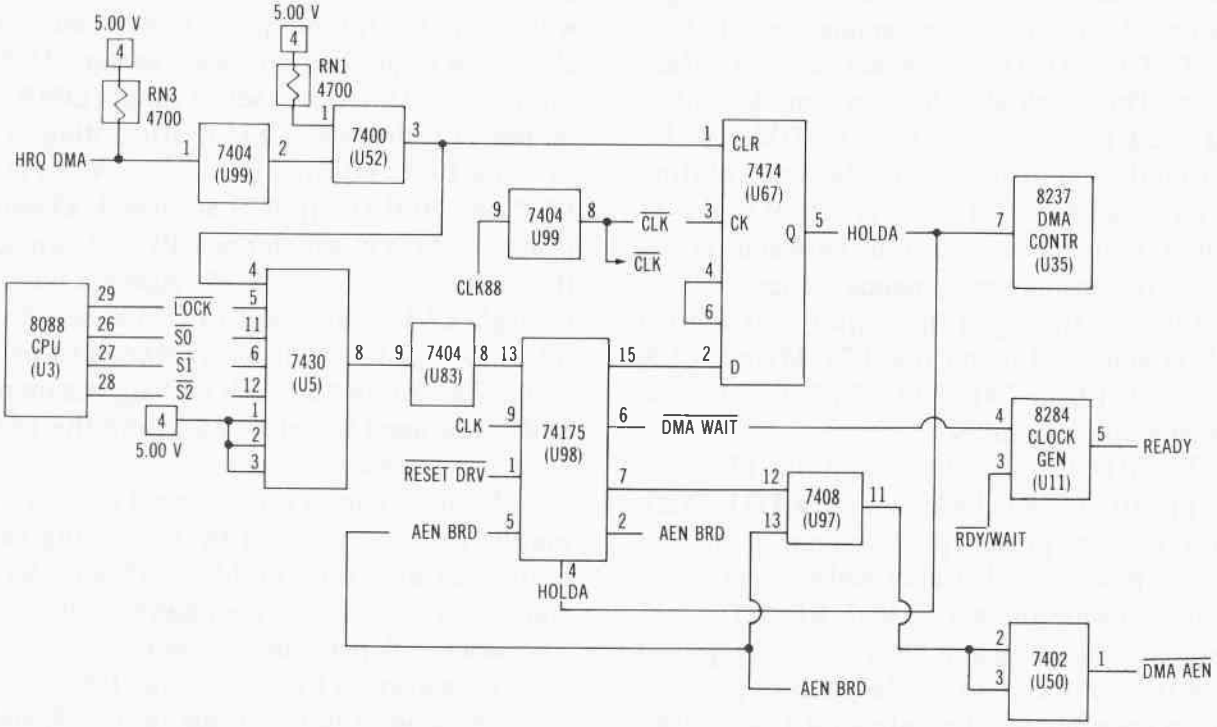


Fig. 2-21. HRQ DMA to HOLDA circuitry.

also becomes one input to 74LS30 eight-input NAND (U5) where it is matched with LOCK*, and the 8088 status signals S0* and S1* from U3 to produce an active high output from pin 8 into pin 9 of hex inverter U83, out pin 8 and into pin 13 (D input) of 74LS175 quad-D latch U98. The Q output from pin 15 of U98 becomes the D input to 74LS74 dual-D latch U67 generating HOLDA out the Q output on pin 5. This causes CPU U3 to tristate its address, data, and control signals giving 8237 U35 control over the address, data, and control buses.

HOLDA feeds back to pin 4, the D input to a latch in U98 generating signal AEN BRD out pin 2. AEN BRD is passed back into the D input (pin 5) of U98 producing (DMA WAIT)* out pin 6. This signal enters pin 4 of 8284 clock generator (U11) to combine with the RDY*/WAIT signal of pin 3 determining the logic state of the READY output from pin 5 into pin 22 of 8088 CPU (U3).

Back to the upper right of Fig. 2-21, HOLDA also feeds out pin 5 of U67 into pin 7 of 8237 U35 indicating that the CPU has relinquished control of the system buses and DMA transfers can proceed.

The first DMA request (DRQ0) comes from the pin 9 Q output of 7474 D latch (U67), shown in Fig. 2-20 and on CF page 57. This signal was developed from the OUT1 signal on pin 13 of the 8253 programmable interval timer U34. Every 72 clock cycles (15 microseconds) programmable interval timer U34 generates OUT1 on pin 13. This signal which clocks a logic high into the D input (pin 12) of U67 producing an active high DRQ0 out pin 9 and input on U35 pin 19.

DRQ0 causes DMAC U35 to produce an output (DACK0 BRD) that causes a memory access that refreshes the charges in the DRAM ICs on the system board. Every 15 microseconds, DMA channel 0 is activated causing a memory access operation. As shown in the upper left of Fig. 2-22, the active high DRQ0 signal in pin 19 causes U35 to generate a (DACK0 BRD)* acknowledge signal out pin 25. This signal is used to clear U67 so another DRQ0 can be generated 15 microseconds later. It also initiates a RAM refresh cycle by activating four row address select (RAS) lines while disabling all memory chip address select (CAS) lines. This also prevents using DRQ0 for anything other than memory refresh.

(DACK0 BRD)* enters pin 13 of 74LS74 U67 to clear this latch and ready it for another OUT1 clock on pin 11. It also goes to pin 6 of 74LS138 decoder U48 where it combines with A18 and A19 to produce RAM address select (RAM ADDR SEL)* out pin 15. (DACK0 BRD)* passes through 74LS244 (U15) to become DACK0* input to the refresh circuitry in the upper right of Fig. 2-22. DACK0* enters pin 1 of 74S00 (U81) to generate (REFRESH GATE)* out pin 3 and to 74LS08 (U49). DACK0* is also passed to pin B19 of expansion slots J1 through J5.

(DACK0 BRD)* also is inverted in 74S04 (U83) to become DACK0, an enable input on pin 4 of 74LS138 decoder U65. Decoder U65 generates the signals that become the active low row address selects (RAS0*, RAS1*, RAS2*, and RAS3*) out pins 3, 6, 8, and 11 of 74LS08 (U49).

Decoder 74LS138 (U47) uses (DACK0 BRD)* on G1 enable input pin 6 to prevent generation of memory column address strobe signals (CAS0*, CAS1*, CAS2*, and CAS3*) out pins 12 through 15). The disabling of all chip address strobe lines ensures that DRQ0 is used only for memory refresh operations.

The other three DMA request channels (DRQ1 through DRQ3) shown in Fig. 2-20 come from pins B18, B6, and B16 respectively on connectors J1 through J5. These request signals cause U35 to generate acknowledge active low signals DACK1*, DACK2*, and DACK3* out pins 24, 14, and 15, as shown in Fig. 2-23. These acknowledge signals tell an individual peripheral that a DMA cycle has been granted. They are used to produce chip select signals to the I/O devices causing the devices to place an input byte on the data bus. The active low DMA acknowledge signal (DACKx through DACK3*) also supplies a memory address and signals the destination to read the address.

Notice that DACK2* and DACK3* are also passed onto pins 4 and 5 of 74LS670 4 x 4 tristate

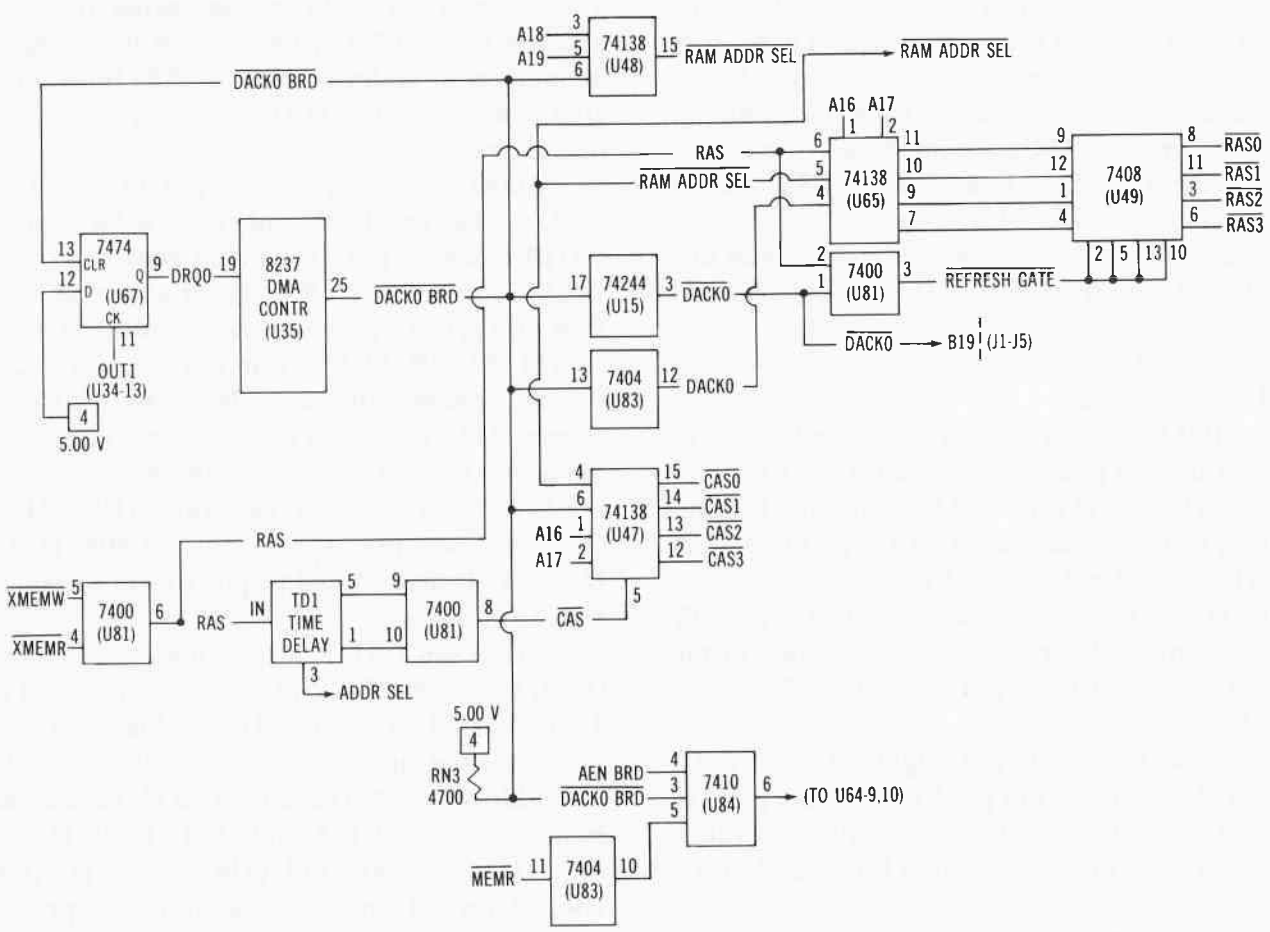


Fig. 2-22. 8237 DRQ0 DMA acknowledge circuitry.

register file (U19). Because DMAC U35 can only handle one 64K memory page at a time, U19 is used as a DMA page register file to increase the DMA capability from 64K to a full one megabyte while retaining the 64K page boundary. U19 is a 16-bit tristate register file organized as four words of 4 bits each. Each word represents a 64K page boundary.

Separate read and write address and enable inputs are available permitting simultaneous reading from one location and writing to another location inside U19. The 4-bit word to be stored is placed on XD0 through XD3 (pins 1, 2, 3, and 15). Write address inputs XA0 and XA1 (pins 13 and 14) determine the location of the stored word. When (WRT DMA PG REG)* on pin 12 goes active low, XD0 through XD3 enter and are latched inside the appropriate word location. This location is transparent to the data while (WRT DMA PG REG)* is low. Data on XD0 through XD3 will be read out in the same logic level as input when the register files are read. The data and write address inputs are inhibited when (WRT DMA PG REG)* is high.

The 64K memory page boundary data that was stored inside U19 is made available on output pins 6, 7, 9, and 10 (A16 through A19) whenever (DMA AEN)* goes active low on input pin 11. This causes the condition of DACK2* and DACK3* to represent an address to one of the four word locations inside U19 copying the data word out as A16 through A19. When (DMA AEN)* returns high, pins 6, 7, 9, and 10 go tristate high impedance. This permits the same address bus segment to use A16 through A19 from the CPU addressing bus circuitry.

Once a DMA request occurs, the DMA active cycle begins. Typically, an active cycle

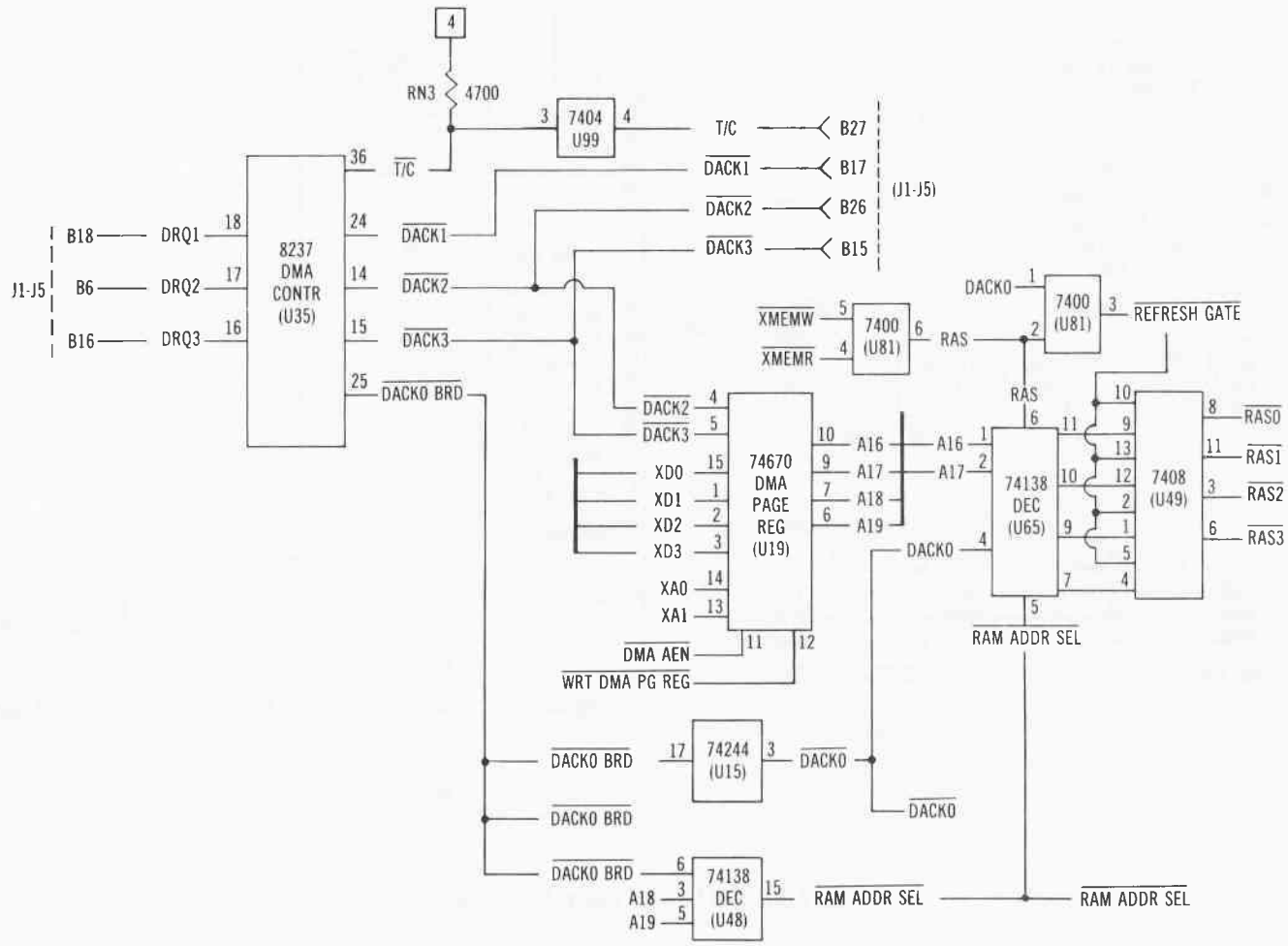


Fig. 2-23. DMA acknowledge circuitry.

requires only five 210 nanosecond clock periods (DMA states), or about 1 microsecond (1.05 μs) of CPU time. This compares quite favorably with the 6 microsecond, 29 clock period typical data transfer time for a CPU memory to memory operation. The memory refresh dummy DMA transfer takes four clocks or 840 nanoseconds.

Figure 2-24 is a snapshot of most of the DMA circuitry. During DMA data transfers, the I/O and memory signals on the lower left are used to control the DMA data direction. If the transfer is from I/O to memory, MEMW*, IOR*, and DACK1*, DACK2*, or DACK3* all go active low. As shown in the lower left of Fig. 2-24, IOR*, IOW*, MEMR* and MEMW* connect to transceiver 74LS245 (U14) to produce XIOR*, XIOW*, XMEMR*, and XMEMW* that connect to 8237 DMAC (U35) on pins 1, 2, 3, and 4. XMEMR* and XMEMW* enter pins 4 and 5 of 74LS00 (U81) and combine to produce the row address strobe (RAS) signal out pin 6 (see Fig. 2-23). To prevent the 8088 CPU buffers (U7, U9, and U10) and the data bus transceiver (U13) from placing addresses and data on the system buses during DMA transfer, the AEN BRD signal from U98 (Fig. 2-21) goes high disabling the outputs of these ICs. (DMA AEN)* from pin 1 of 74LS02 NOR (U50) shown in the lower right of Fig. 2-21 and CF page 57 is brought active low enabling the DMA buffers and latches, and DMA page register permitting high speed data transfers.

From the lower right of Fig. 2-19, you will notice that the 8237 data bus is bidirectional (XD0 through XD7). This data path is multiplexed with the high address byte (A8 through A15) using 74LS373 latch buffer (U18). The XD0 through XD7 output from U35 is

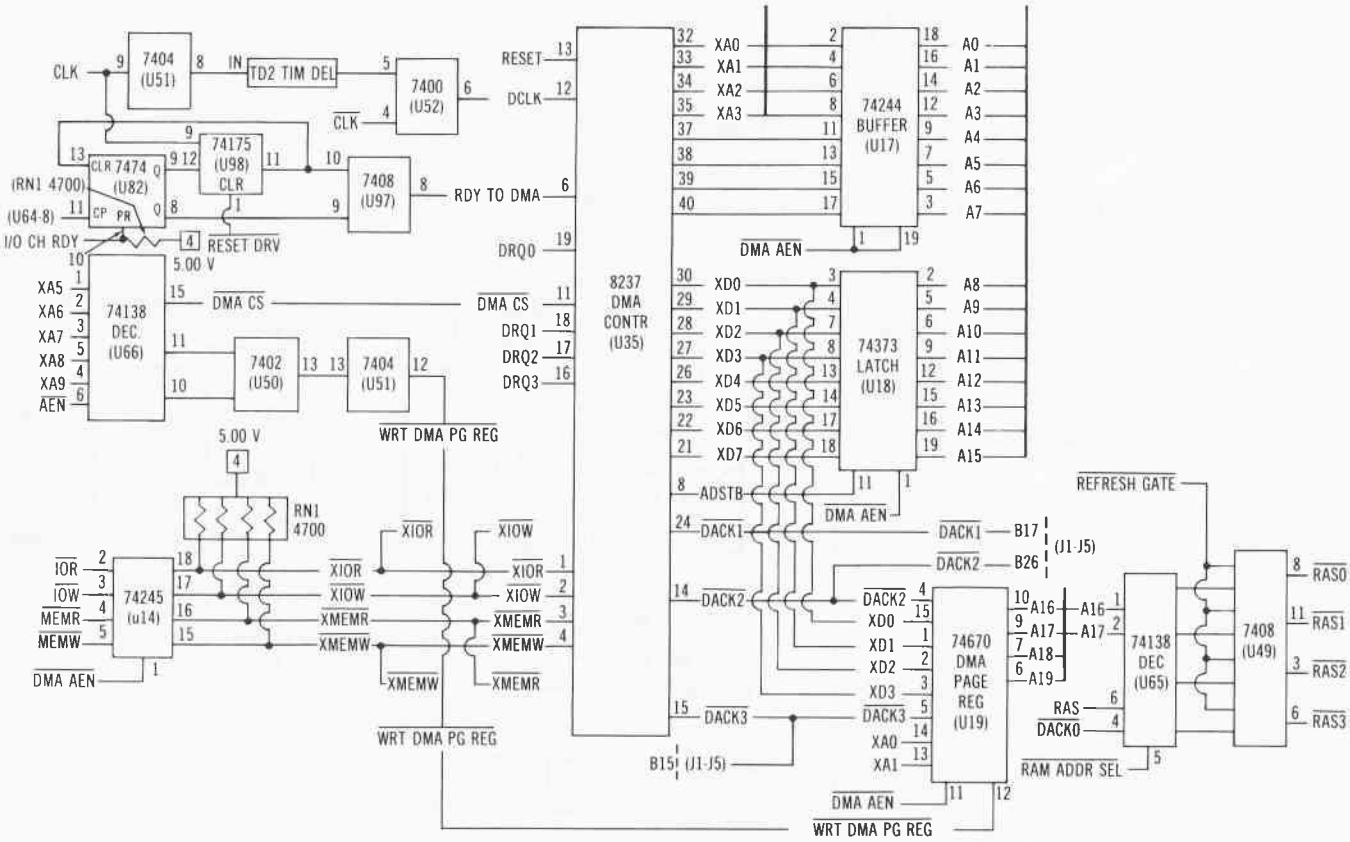


Fig. 2-24. 8237 DMA controller circuitry.

latched in U18 by the address strobe signal (ADSTB) out pin 8 of U35 into enable input (pin 11) of U18. In addition, the normally input lines XIOR*, XIOW* and XA0 through XA3 are actually bidirectional and become output lines during a DMA transfer.

During an active DMA cycle, if more time is required (accessing a slow memory or I/O device), a ready signal (RDY TO DMA) is input to U35 by 74S08 NAND U97 pin 8 inserting wait states in the DMA cycle as shown in Fig. 2-25.

Logic high inputs from inactive (DACK0 BRD)*, XMEMR output of 74LS04 (U83), and AEN BRD enter 74LS10 three-input NAND (U84) produce two inputs to 74LS20 four-input NAND (U64). The other two inputs to U64 are XIOR* and XIOW* on pins 13 and 12 respectively. The pin 8 output from U64 enters the clock input (pin 11) of 74LS74 dual-D latch (U82) latching the condition of I/O CH RDY from expansion connections A10 of J1 through J5 into U82. The Q output from U82 (pin 9) is labelled RDY*/WAIT. This signal is clocked into the D input (pin 12) of 74LS175 quad-D latch (U98) by the CLK signal on U98 pin 9. It is also passed to pin 3 of 8284 clock generator U11 where it is combined with (DMA WAIT)* and used to control the READY signal out pin 5 of U11.

Terminal Count (T/C)

Each time the 8237 DMA internal word count circuitry reaches a count of FFFFH a terminal count signal is generated causing U35 pin 36 to go active low. As shown in Fig. 2-26, the (T/C)* signal out pin 36 is passed through 74LS04 Hex inverter (U99) to produce an active high T/C output signal to pin B27 of the expansion connections J1 through J5. Signal T/C is used to indicate completion of a DMA operation.

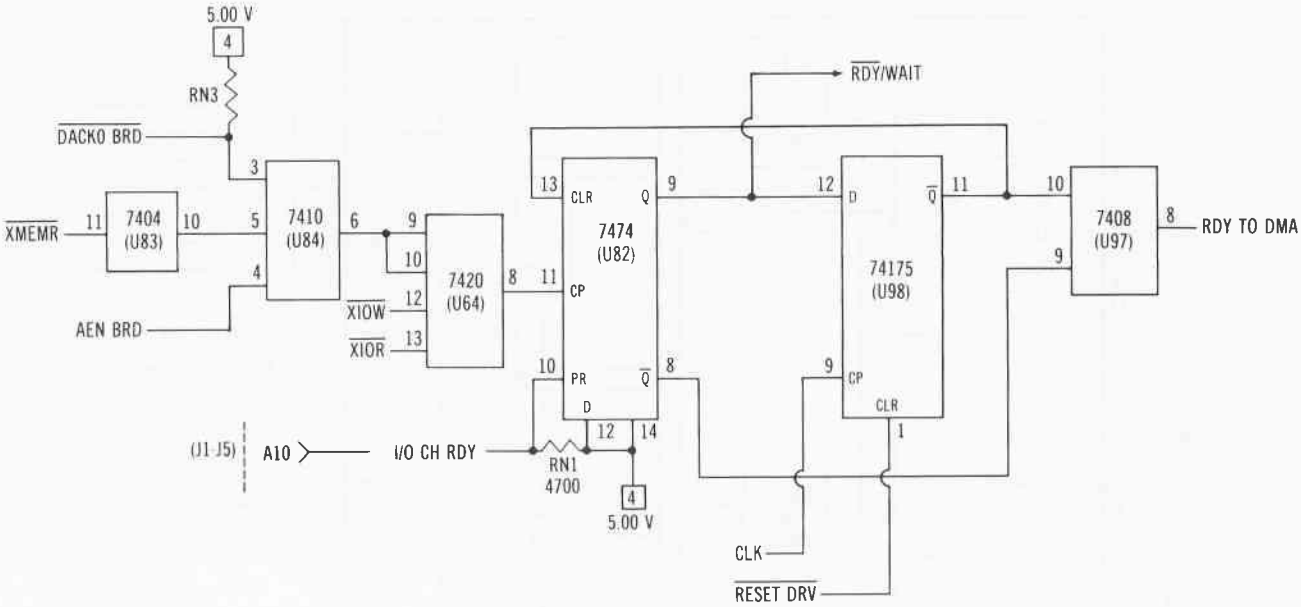


Fig. 2-25. RDY TO DMA circuitry.

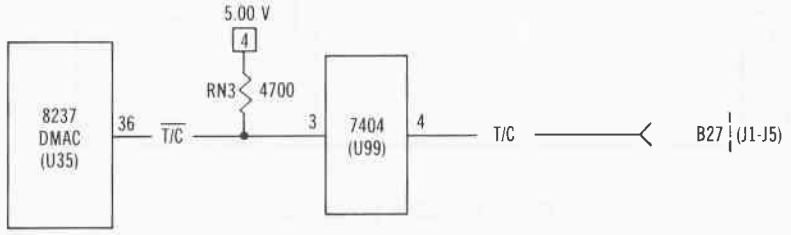


Fig. 2-26. Terminal count circuitry.

T/C should not be confused with the timer controller chip select (T/C CS)* signal out pin 13 of 74LS138 decoder (U66) to the input chip select (pin 21) of 8253 Programmable Interval Timer (U34) shown in Fig. 2-18.

8255 PROGRAMMABLE PERIPHERAL INTERFACE (U36)

One of the most important interconnects on the IBM PC system board is a 40-pin IC called the 8255 programmable peripheral interface (U36). U36 is used to interface peripheral equipment to the 8088 system bus. Figure 2-27 shows that U36 interfaces a bidirectional data bus with three programmable I/O ports under control of specialized read/write control logic. Figure 2-27 is drawn as U36 is configured in your IBM PC.

The functional configuration of U36 is programmed during initialization. The tristate bidirectional buffered data bus XD0 through XD7 connects with U36 on pins 27 through 34. Data is transferred by the buffer under CPU control. PPI control words and status information are also passed back and forth through this data bus buffer.

The read/write control logic accepts six input control signals that manage the operation of this device. An active low XIOR* on pin 5 enables U36 to send data or status information to the CPU (U3) over the data bus. It lets U3 read from U36. An active low XIOW* on pin 36 enables the CPU to write data or control words into U36. XA0 and XA1 on pins 9 and 8 respectively are used with XIOR* and XIOW* to select one of the three I/O ports or internal word control registers. Typically, these four signals are

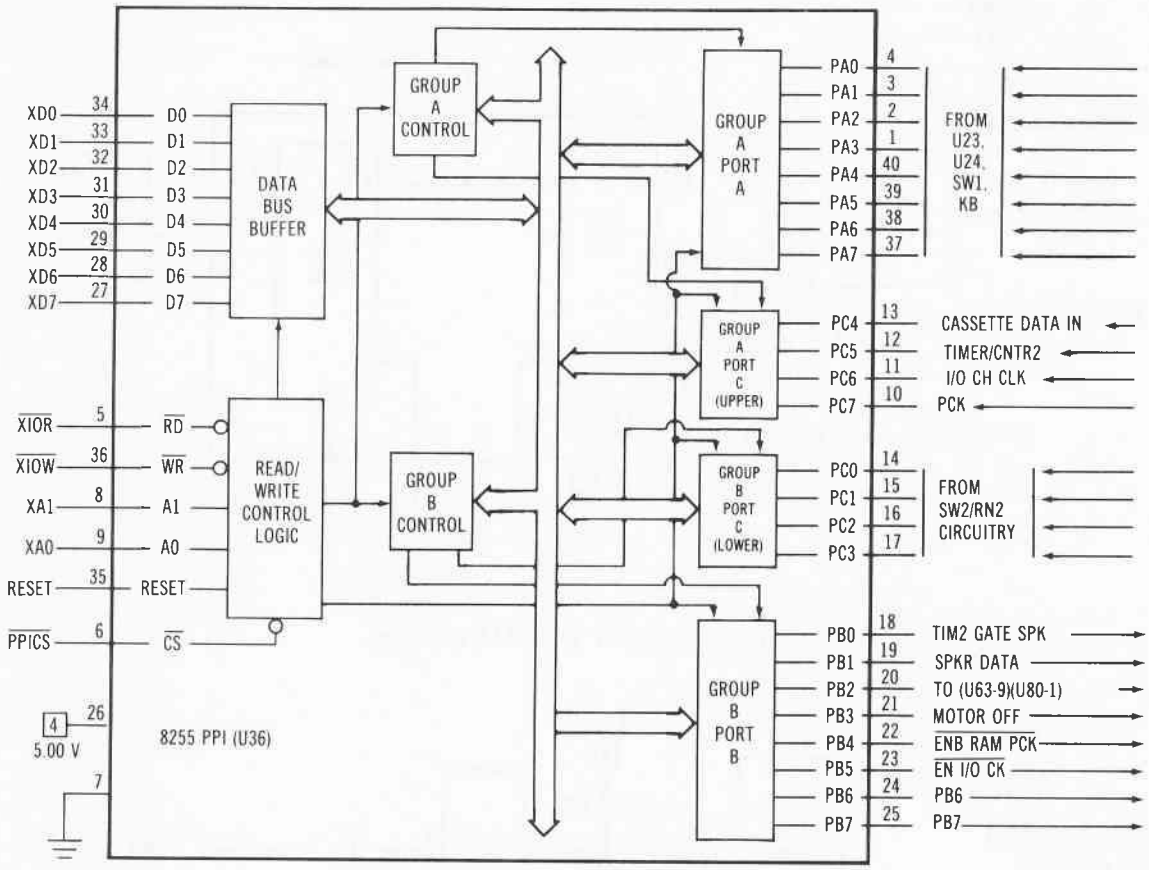


Fig. 2-27. 8255 programmable peripheral interface.

used to cause port A or port C to write to the data bus, and the data bus to pass information out port B. An active high on RESET input pin 35 clears the control register and sets all ports to the input mode. This prevents input circuitry to ports A and C from being damaged if one of these input ports become an output port after initialization. Finally, the active low programmable peripheral interface chip select (PPICS)* signal on pin 6 is used to enable communication between U36 and the CPU (U3).

The Group A and B control blocks accept commands from the read/write control logic, and control words from the data bus buffer to generate proper configuration commands to the I/O ports. Group A controls port A and the upper nibble of port C. Group B controls port B and the lower nibble of port C.

The three 8-bit I/O ports (A, B, and C) can be configured in a wide variety of functional ways, but are configured by design and software to be specific input or output. The port configuration is determined by a control word copied into U36 via the data bus. This control byte (10011001) configures ports A and C as inputs and port B as an output. This basic input/output configuration requires no handshaking for information transfer. Once configuration is complete, a control input to the read/write control logic causes U36 to operate according to an input code on pins 5, 8, 9, and 36 (XIOR*, XA1, XA0, and XIOW*). The chip control signal configuration for operation of each port is shown in Table 2-3.

U36 operates asynchronously. Once the correct code is present and the chip is selected (pin 6), the data transfer direction is predetermined. Figure 2-28 shows that port A is connected to the 74LS322 keyboard bidirectional register (U24) and the 74LS244 buffer (U23).

Table 2-3. Control Logic Input Configuration for Port Operation

A1	A0	XIOR*	XIOW*	FUNCTION
0	0	0	0	Port A to data bus
1	0	0	1	Port C to data bus
0	1	1	0	Data bus to port B
1	1	1	0	Data bus to control logic
X	X	1	1	Data bus to tristate
X	X	X	X	Data bus to tristate

Depending on the status of output port B bit 7 (PB7), port A reads the keyboard or the system board switch (SW1). PB7 high causes port A to read the keyboard input byte. PB7 low causes port A to read the system board DIP switch (SW1).

Port C receives inputs from an attached cassette peripheral, the 8253 programmable interval timer (U34) counter 2 output, the status of the I/O channel, the status of RAM parity checking, and the number of RAM banks in the I/O channel as determined by the position of the memory configuration switch (SW2). This lets the PC recognize up to eight 64K byte memory banks in the I/O channel.

Port B outputs information to the speaker, to the timer (U34), to the cassette motor, and to the parity test circuitry. When PB0 is high, coun-

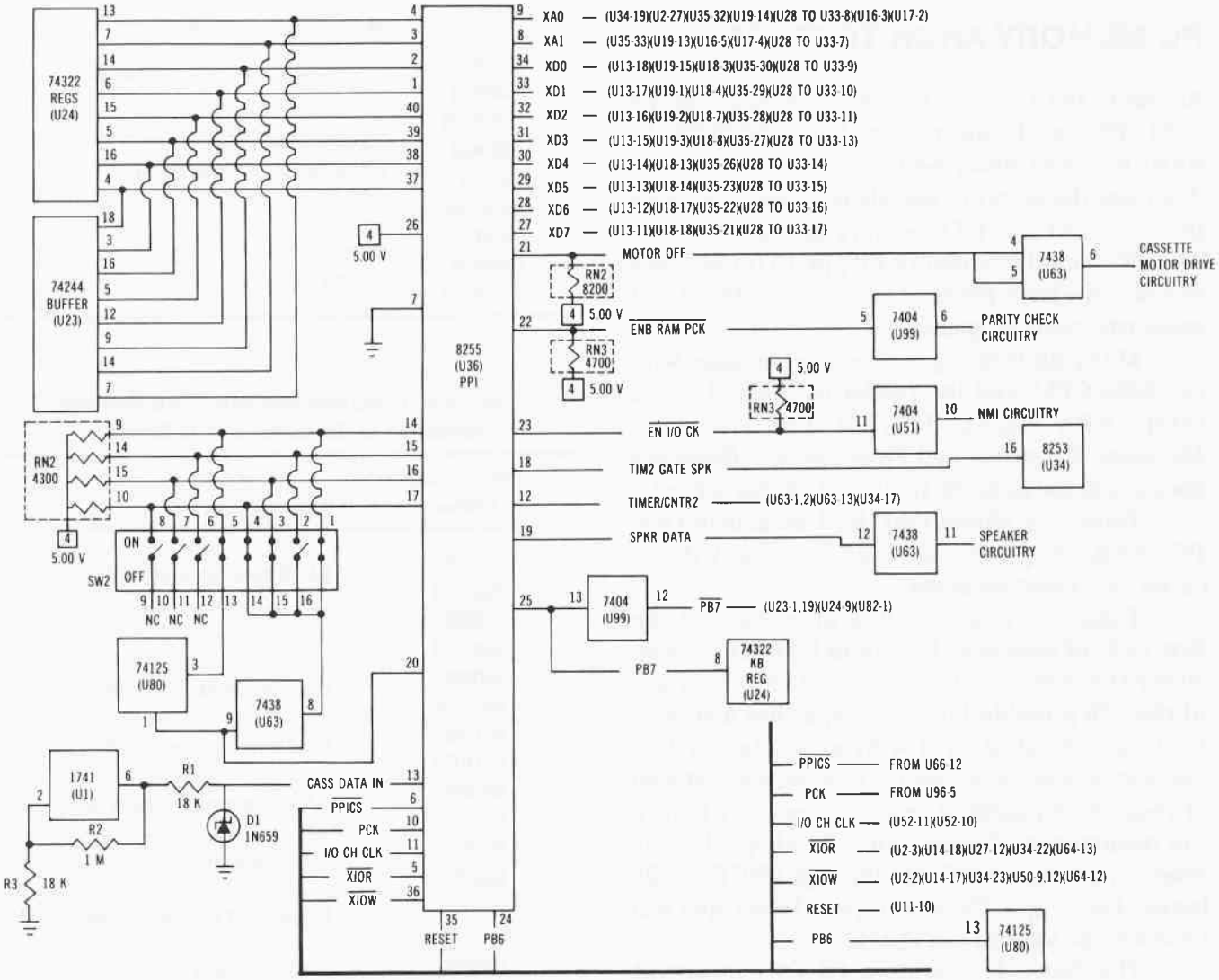


Fig. 2-28. Circuitry associated with 8255 PPI U36.

ter timer 2 in U34 generates a square wave for the speaker and cassette outputs. PB1 is used to send programmed data waveforms to the speaker. PB2 high is passed to 74LS125 quad tristate buffer (U80) and 74LS38 quad two-input NAND (U63) to enable the low nibble of the system board switch SW2 (see Fig. 2-28). When PB3 goes high, the cassette motor is turned off. If PB4 and PB5 are low, RAM parity and the I/O channel status lines read by port C are enabled. Port B PB6 is used to drive the keyboard clock line low (PB6 = 0), and PB7 is used to cause port A to read SW1 (PB7 = 1) or the keyboard data (PB7 = 0).

PC MEMORY ARCHITECTURE

Access to the 1 million byte memory space of the IBM PC, with addresses from 00000H to FFFFFH, was described earlier. This section describes the circuitry and signals associated with the on-board and I/O memory in your machine. The PC includes memory-mapped I/O because devices can have physical addresses within the 1 megabyte memory space.

Many addressing modes are possible with the 8088 CPU, and the reader is referred to the Intel iAPX 86/88, 186/188 *User's Manual Hardware Reference* and *Programmer's Reference* documents for in-depth treatment of this subject.

Table 2-4 shows that the 1 megabyte IBM PC memory space is divided into many dedicated and reserved areas.

Table 2-5 describes the allocation of the first 64K of memory. Locations 00000H through 003FFH are reserved for interrupt vectors. Each of the 256 possible interrupt types has a service routine indicated by a 4-byte pointer in this memory space. The pointer elements are stored at respective places in reserved memory before the occurrence of interrupts. Intel specifically reserves addresses 00000H through 0007FH (128 bytes) for 32 specific interrupts. Interrupts will be discussed later in this chapter.

The basic I/O system (BIOS) interrupt vectors 00 through 1F are contained in addresses

Table 2-4. Memory Allocation for the IBM PC

Hexadecimal Address	16K/64K System	64K/256K System
00000H 0FFFFH	64K RAM on system board	256K RAM on system board
10000H 3FFFFH	576K RAM on expansion cards	
40000H 9FFFFH		384K RAM on expansion cards
A0000H AFFFFH	(A0000H - BFFFFH = 128K reserved for displays)	
B0000H B3FFFH	Monochrome video RAM	
B4000H B7FFFH		
B8000H BBFFFH	Color/graphics video RAM	
BC000H BFFFFH		
C0000H FFFFFH	ROM	

Table 2-5. (16K/64K and 64K/256K Systems) Allocation of the Lower 64K of Memory

Hexadecimal Address	Memory Content
00000H 0001FH	BIOS interrupt vectors 00-1F
00020H 0007FH	
00080H 0009FH	DOS interrupt vectors 20-3F
00100H 001FFH	User interrupt vectors 40-7F
00200H 003FFH	BASIC interrupt vectors 80-FF
00400H 004FFH	BIOS data area
00500H 005FFH	BASIC and DOS data area
00600H 0FFFFH	63.5K user area

00000H through 0001FH. The next set of vectors (DOS interrupt vectors 20 through 3F) are located between 00080H and 0009FH. USER interrupt vectors 40 through 7F reside in locations 00100H through 001FFH. Locations 00200H through 003FFH contain BASIC interrupt vectors 80 through FF. Special BIOS data is stored between 00400H and 004FFH. Then BASIC and DOS data are stored between 00500H and 005FFH. From location 00600H to 0FFFFH is 63.5K of user defined RAM memory space.

Read-only memory (ROM) is designed to reside in the address space bounded by C0000H and FFFFFH. The allocation of these 262,144 locations (256K memory space) in both the 16K/64K and 64K/256K PC systems is shown in Table 2-6.

Table 2-6. ROM Memory Space Allocation

Hexadecimal Address	System Board Chip	16K/64K Systems	64K/256K Systems
C0000H C7FFFH	-	(32K)	192K ROM for system expansion and control
C8000H CBFFFH	-	(16K) Hard disk control	
CC000H EFFFFH	-	(144K)	
F0000H F3FFFH	-	(16K)	
F4000H F5FFFH	U28	(8K)	open ROM socket
F6000H F7FFFH	U29	(8K)	32 ROM CASSETTE BASIC
F8000H F9FFFH	U30	(8K)	
FA000H FBFFFH	U31	(8K)	
FC000H FDFFFH	U32	(8K)	
FE000H FFFFFH	U33	(8K)	ROM BIOS

Memory locations FFFF0H through FFFFFH (16 bytes) are reserved for specific 8088 CPU operations including a jump to the initial program loading routine. When a RESET occurs, the CPU always begins executions at system reset location FFFF0H where a jump instruction must be stored shifting program execution to a special routine.

The I/O space in the IBM PC is separate from the memory space, although I/O devices can also be configured to be addressable within the memory space (memory mapped I/O). The I/O memory space accessible by the 8088 CPU covers addresses 0000H to FFFFH (64K bytes). The allocation of I/O memory is shown in Table 2-7.

Table 2-7. IBM PC I/O Address Map

Hexadecimal Address	Access Function
000-00F	8237 DMA controller (U35)
020-021	8259 programmable interrupt controller (U2)
040-043	8253 programmable interval timer (U34)
060-063	8255 programmable peripheral interface (U36)
080-083	DMA page registers
0A	NMI mask register enable/disable
0C	Reserved
0E	Reserved
0F8-0FF	Reserved for 8088 upgrades
100-1FF	Not usable
200-20F	Game control
210-217	Expansion unit
220-24F	Reserved
278-27F	Reserved
2F0-2F7	Reserved
2F8-2FF	Asynchronous communications (secondary)
300-31F	Prototype card
320-32F	Hard disk
378-37F	Printer
380-38C	Synchronous data link control (SDLC) communications
380-389	Binary synchronous communications (secondary)
3A0-3A9	Binary synchronous communications (primary)
3B0-3BF	IBM monochrome display/printer card
3C0-3CF	Reserved
3D0-3DF	Color/graphics
3E0-3F7	Reserved
3F0-3F7	Diskette
3F8-3FF	Asynchronous communications (primary)

Table 2-7 shows that some of the chips on the system board are directly accessible via an I/O write or read command. Other locations in I/O memory space are reserved by IBM or a chip manufacturer. Intel reserves addresses 000F8H through 000FFH (8 bytes) for future hardware and software products.

The 8088 accesses memory in bytes. Word operands are accessed in two bus cycles. Instructions are also fetched 1 byte at a time.

MEMORY AND I/O ACCESS AND CONTROL SIGNALS

The circuitry that generates the four memory command signals from 8288 bus controller U6 to enable memory read and write operations is shown in Fig. 2-29. Each of the active low signals–memory read (MEMR*), memory write (MEMW*), I/O read (IOR*), and I/O write (IOW*)–originate in the 8288 bus controller (U6). Figure 2-29 shows that these four signals are passed to 74LS245 transceiver (U14) where they become the XMEMR*, XMEMW*, XIOR*, and XIOW* used throughout the system board. They are also passed to pins B11 through B13 on expansion connectors J1 through J5 for use on peripheral boards.

The logic value combination of the CPU status bits S0*, S1*, and S2* forms a code that determines what system operation is to occur. Table 2-8 repeats the Table 2-2 description of the operation that is enabled by each combination of the status code inputs, but includes the pin assignments for each specific memory or I/O command generated.

Therefore, whenever the status bit input into 8288 (U6) in Fig. 2-29 is 001, 010, 101, or 110, an appropriate active low memory or I/O command is generated on pin 7, 8, 12, or 13. These commands are felt on the inputs to 74245 transceiver-buffer (U14) and on pins B11 through B14 of the expansion sockets J1 through J5. On the down side of transceiver U14, the memory and I/O commands assume a new label.

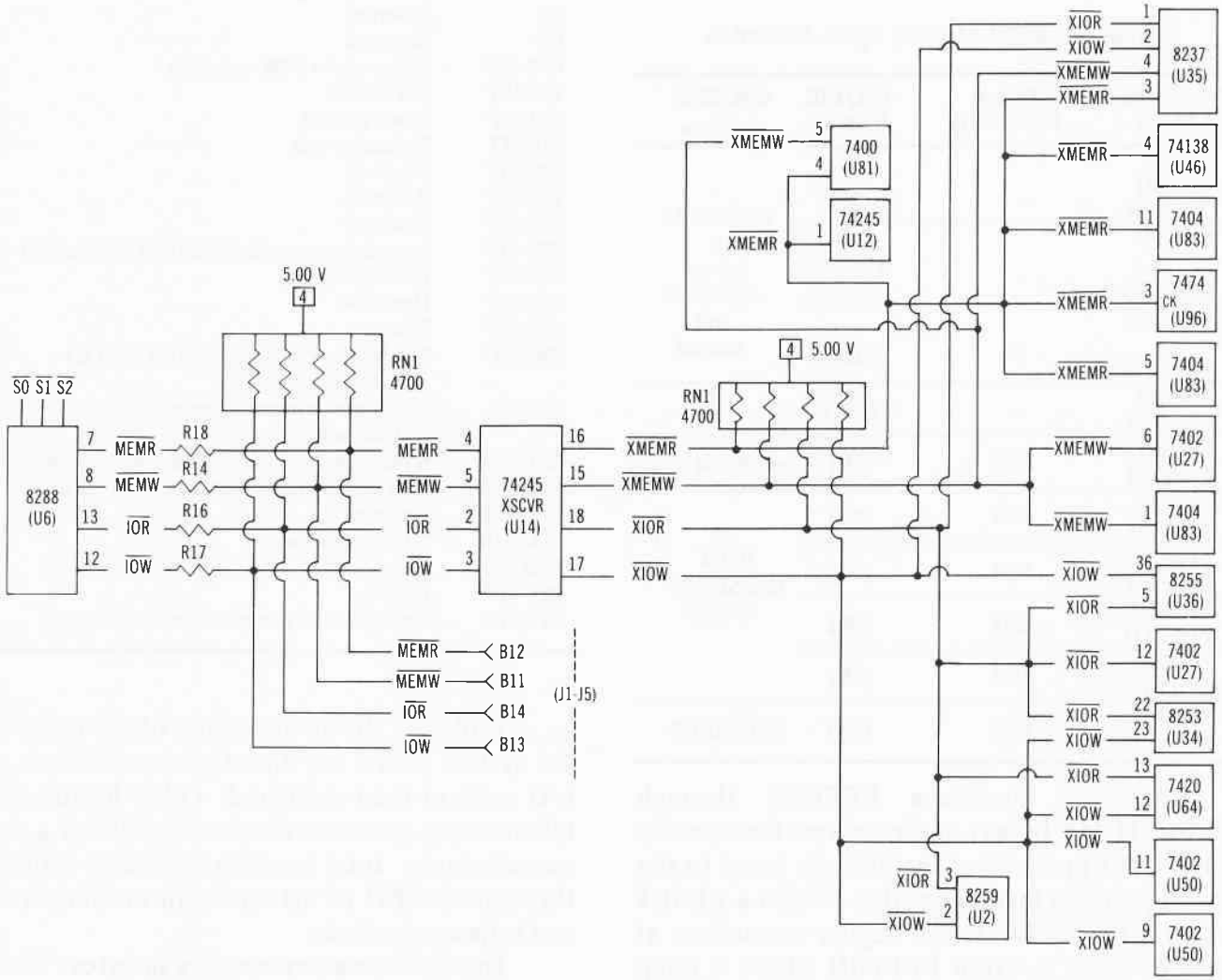


Fig. 2-29. Memory and I/O bus command signals.

Table 2-8. Status Bit Code Allocations

Status Bits S2*	S1*	SO*	Processor State	8288	Command Pin #
0	0	0	Interrupt acknowledge	INTA*	Pin 14
0	0	1	Read I/O port	IOR*	Pin 13
0	1	0	Write I/O port	IOW*	Pin 12
0	1	1	Halt	None	
1	0	0	Code access	MEMR*	Pin 7
1	0	1	Read memory	MEMR*	Pin 7
1	1	0	Write memory	MEMW*	Pin 8
1	1	1	Passive	None	

Memory read (MEMR*) becomes active low buffered memory read (XMEMR*). Memory write (MEMW*) becomes active low buffered memory write (XMEMW*). I/O read (IOR*) becomes active low buffered I/O read (XIOR*). And I/O write (IOW*) becomes active low buffered I/O Write (XIOW*). These signals are passed throughout the system board, as shown in Fig. 2-29. They are typically connected as enable inputs to gates used in memory or I/O operations. In the figures to follow in this section, you will note one or more of these signals used in every memory or I/O operation.

READ ONLY MEMORY (ROM)

The system board in every PC is designed to hold six 8K x 8-bit ROM chips–only five are installed. Figure 2-30 shows the pinout allocation for the 2364 ROM chips installed in your system board. Twelve address lines connect to each chip. A special chip select signal on pin 20 enables that particular ROM so 8 bits of data can be placed on the output pins 9 through 11, and pins 13 through 17.

Of the assigned ROM memory space (C0000H to FFFFFH), locations C0000H to F3FFFH are reserved for future ROM code. If a hard disk is connected, its allocated address space is from C8000H to CBFFFH. Address space F4000H to F5FFFH was reserved for part of the original cassette BASIC code. These addresses are allocated to the single empty socket (U28). The next four ROM chips (U29

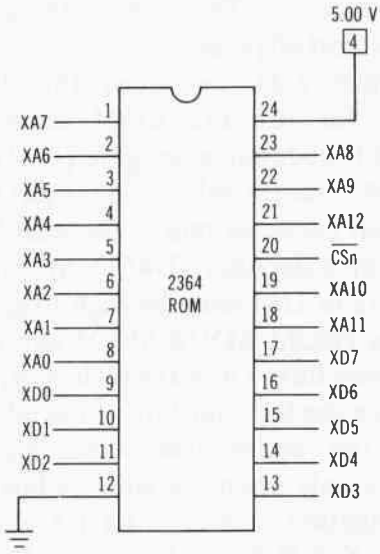


Fig. 2-30. IBM PC ROM chip pin assignment.

through U32), are allocated address space (F6000H through FDFFFH) and contain the 32K cassette BASIC object code. The remaining ROM (U33) has allocated address space FE000H to FFFFFH and contains the PC's basic input/output system (BIOS) routines.

Permanently stored in the 8K ROM BIOS chip (U33) are program codes called "I/O drivers" that enable printer operations and dot patterns for 128 characters to enable graphics display operations. Inside U33 there is also code to control the time of day clock, cassette operations, the power-up self test, and minifloppy disk bootstrap loading. All address locations below C0000H are dedicated for I/O and random access memory (RAM).

The internal ROM-based PC power-up self test occupies 2K of the 8K BIOS ROM and includes a series of 14 tests that occur each time power is applied. These tests check the 8088 CPU (U3), the keyboard, the video display adapter card, the cassette recorder, the floppy disk interface, and the ROM and RAM. The RAM test contains five different write/read operations over the entire RAM memory. Each memory test writes and then reads and checks a different bit pattern in memory. The RAM tests can take as long as 1.5 minutes if your system board is fully populated. The power-up self tests are bypassed if the computer is restarted with

power already on. This reduces the initialization time by almost 40 percent.

Figure 2-31 describes the chip select circuitry for the PC ROM memory. The 74LS20N four-input NAND gate (U64) generates the ROM address select signal (ROM ADDR SEL)* that becomes one of the enable inputs to 74138 1-of-8 decoder (U46). All four address line inputs to U64 must be high to generate the active low (ROM ADDR SEL)* output on pin 6. The address lines (A16 through A19) connected to U64 are the top four bits of the address word, causing the active low select signal to be generated only when the address bus contains a bit combination such as: 1 1 1 1 X X X X X X X X X X X X X X X X, placing the ROM enable signal at F0000H or higher.

Decoder U46 has two more enable signals attached: XMEMR* from the circuitry in Fig. 2-29 and (RESET DRV)* from the circuitry in Fig. 2-7. The latter signal is attached to an active high enable input pin, so only when (RESET DRV)* is high will this combination of enables cause U46 to produce an output determined by the address inputs (A13 through A15) on pins 1 through 3. Once enabled, U46 will produce a low on an output line determined by the code on input pins 1 through 3.

Notice that not all outputs are used on U46. The pins 14 and 15 outputs are unused. This means that one of the remaining six outputs (CS2* through CS7*) will become active low whenever A13 through A15 is a combination of 010 to 111. Connecting this combination to our input combination to U64, we discover that a ROM chip will be selected (given the correct enable inputs to the select circuitry) whenever the address bus code is 1 1 1 1 0 1 0 X X X X X X X X X X X X X X X X or an address of F4000H or higher. This correlates nicely with the F4000H lowest address allocated to the empty ROM socket U28.

The CS7* line goes active low whenever A13 through A15 are all high. This equates to an address code of F E X X X H or higher. CS7* enables ROM U33, whose address space is FE000H to FFFFFH. In all ROM enables, the particular ROM that is selected places an 8-bit

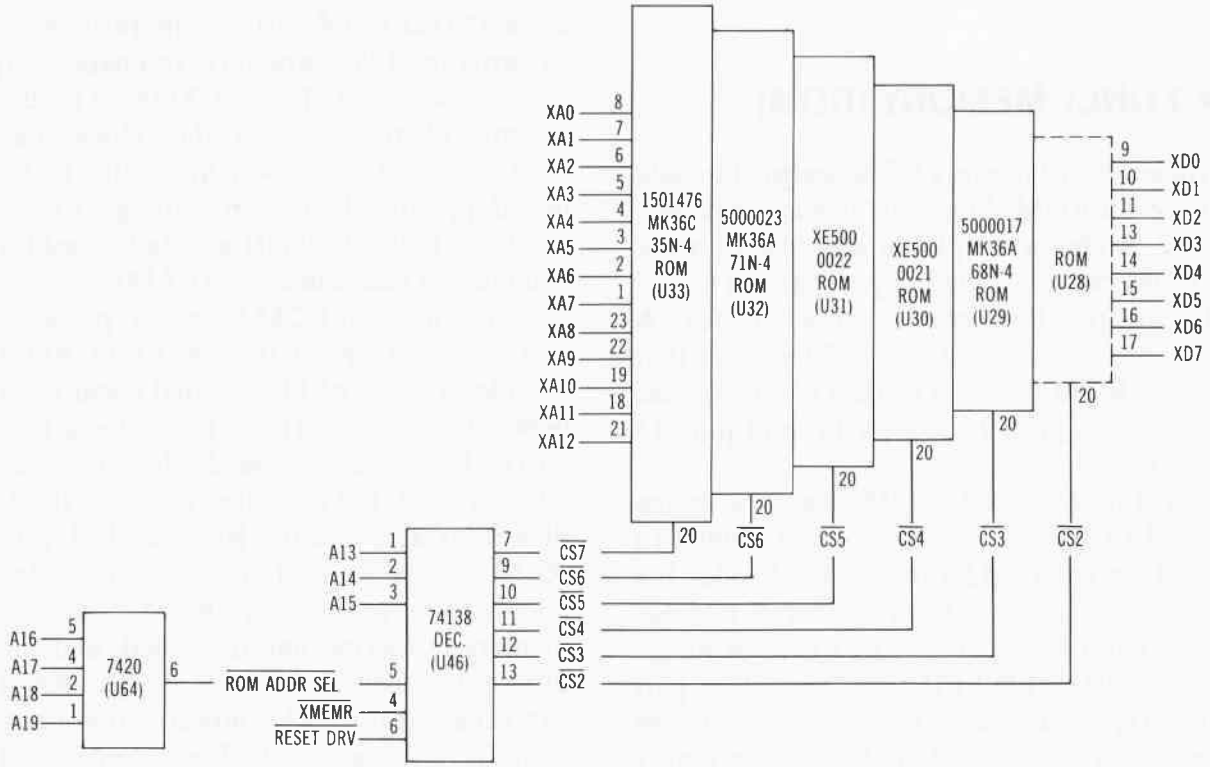


Fig. 2-31. ROM select circuitry.

data word out pins 9 through 17 (XD0 through XD7) determined by the combination of the remaining 12 address bits (XA0 through XA12) on the input (left) side of the memory array in Fig. 2-31.

Figure 2-32 is an example showing the logic levels on the inputs and outputs of the ROM memory circuitry when the address location FE000H is placed on the address bus. This equates to the first address in BIOS ROM U33.

The first 13 address bits are buffered through two 74LS244 buffer chips (U16 and U15) to produce buffered address bits XA0 through XA12. The top 7 bits of the address bus (A13 through A19) are used to generate the appropriate active low chip select signal (in this case CS7*). With the address bus combination shown in the figure, the data stored in location FE000H in U33 is placed on the buffered data bus (XD0 through XD7) between 27 to 39 nanoseconds after CS7* goes low. Access time for ROM is 250 nanoseconds. A memory cycle time is 375 nanoseconds.

The buffered data out of the ROM passes through 74LS245 transceiver U13 to become the data bus (D0 through D7) input for 8088 CPU U3. Figure 2-33 shows how data from a ROM reaches the CPU. When a location is addressed and chip select becomes active low, 8 bits of data become valid on ROM output lines XD0 through XD7 (pins 9 through 11, and 13 through 17). This data is felt on pins 11 through 18 of 74LS245 transceiver U13. Direction signal DIR and enable signal AEN BRD are both low causing

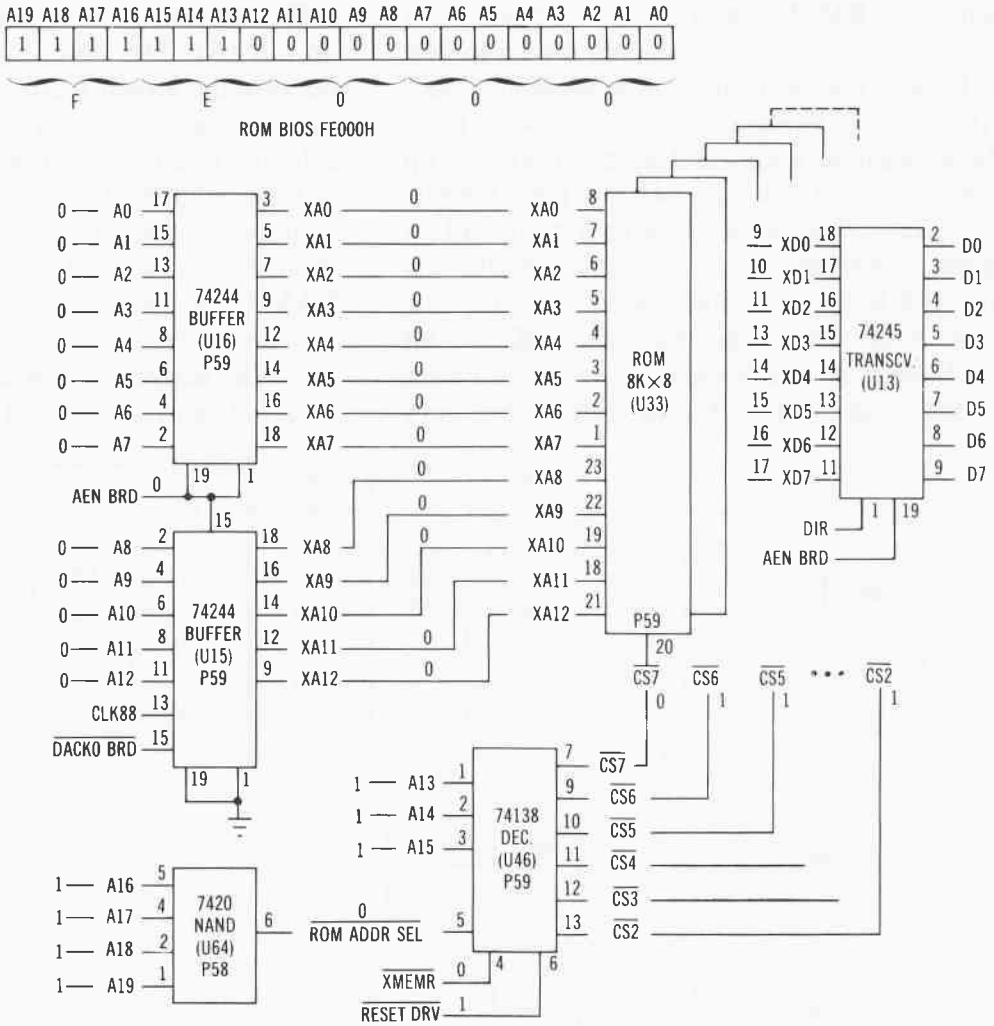


Fig. 2-32. ROM memory circuitry. Accessing the first location in BIOS ROM (U33).

U13 to pass data from the buffered data side (XD0 through XD7) to the data bus side (D0 through D7) on pins 2 through 9.

The output of U13 is passed into 74LS245 transceiver U8 on pins 11 through 18. Data transmit/receive signal DT/R* is active low (the receive condition) as is enable signal G*. This causes U8 to pass data in from pins 11 through 18 and out pins 2 through 9. The output of U8 is the address data bus (AD0 through AD7) which connects directly into 8088 CPU U3 on pins 9 through 16. In this manner, data from any ROM chip can be read into the CPU for action.

RANDOM ACCESS MEMORY (RAM)

When the IBM PC was first introduced, it contained system boards designed for up to 64K of 4116-type 16K by 1-bit random access memory (RAM) chips. These system boards were later replaced with an updated design containing up to 256K of the 4164-type 64K-by-1-bit RAMs. The expansion slots enable a user to install an additional 384K bytes of external RAM allowing up to 640K bytes of available RAM memory in a system. Most users today have the 64K-by-1-bit RAM boards, so this book focuses on the newer system boards. Older board designs are very similar to the system boards described in this book, so you should not have any difficulty following this text and relating the information to your system.

The RAM chips on the system board are mounted in four rows, or banks, of nine chips each. The ninth chip is used to store a parity bit associated with each 8-bit word in that row of RAM. All of the RAM is parity checked during readout. This operation will be described later.

Figure 2-34 describes the pin assignments for a typical MK4564 64K-by-1-bit RAM chip. The RAM access time is 250 nanoseconds, and a memory cycle time is about 410 nanoseconds.

Figure 2-35 is a block diagram of the inside of a 64K-by-1-bit RAM chip. Note that each chip has only eight address lines (MA0 through MA7) connected to it, with a single data input pin (MDn on pin 2), and a single data out line (MDn on pin 14). Only eight address lines are used because the memory array is comprised of 65,536 dynamic memory cells accessed by row and by column. In a memory operation, the row address is entered into the chip first, followed by a column address. Three active low control signal input lines are used to select and enable each RAM chip: row address select (RAS*), column address select (CAS*), and write enable (WE*).

The active low row address strobe (RASn*) on pin 4 is used to begin a memory cycle. The

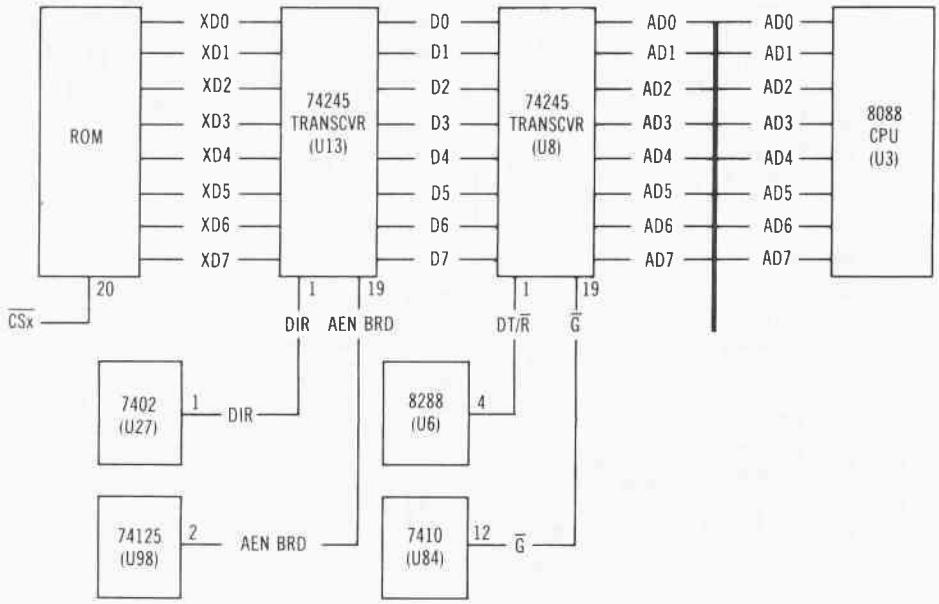


Fig. 2-33. Data transfer from ROM to CPU.

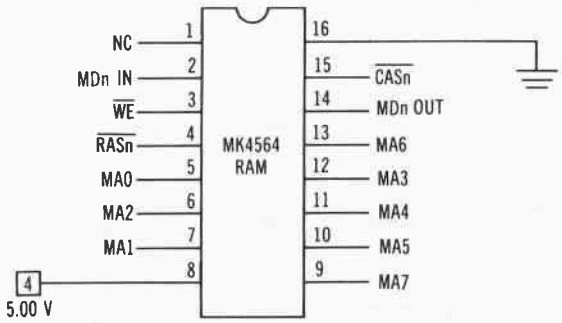


Fig. 2-34. IBM PC RAM chip pin assignments.

"n" suffix identifies a particular bit in a word. For example, RAS5* refers to bit 5 in an 8-bit byte. RASn* allows the row address bits to enter and be latched in the row address buffer and enables the activation of the memory row indicated by the address input. RAS* is also held low while strobing each of the row addresses in a row refresh operation.

The active low column address strobe (CASn*) on pin 15 is used to latch eight column address bits into the column address buffer. It also activates the column in the memory array decoded by the input memory address bits.

The active low write enable (WE*) on pin 3 is used to put the chip into a write or read mode. When WE* is high, the read mode is selected. An active low WE* input causes the chip to enter the write mode.

RAM MEMORY OPERATION

Approximately 500 microseconds after power is first applied to the system, and after RAS* has been held high for about 100 microseconds, the RAM circuitry is initialized by a software routine that does eight RAS* cycles. This initializes the dynamic RAM for the next memory activity.

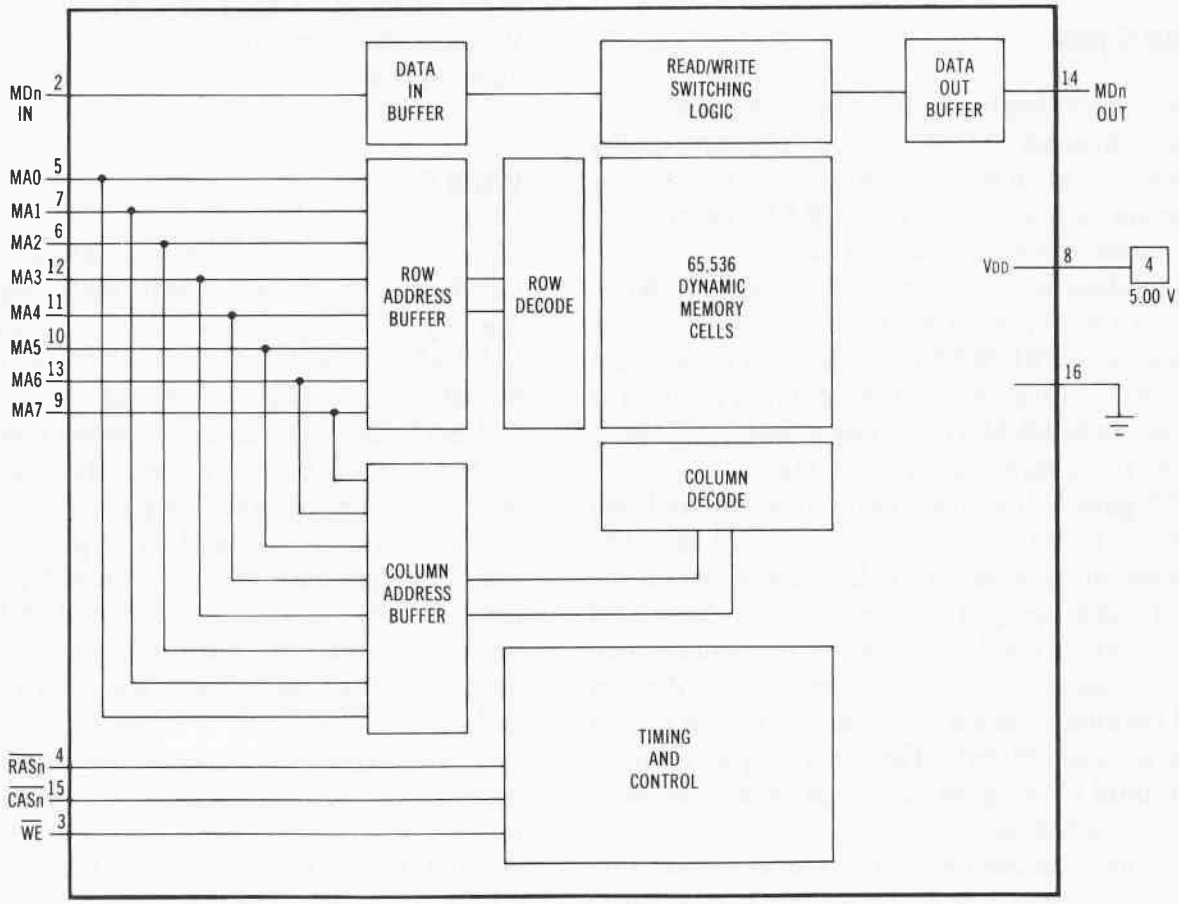


Fig. 2-35. Block diagram of 64K dynamic RAM.

Sixteen address bits are needed to decode one of 65,536 storage cell locations in a single RAM chip. Because each chip has only 8 address bit inputs, the 16 addresses needed to access each memory cell are multiplexed into the chip, row address first, followed by a column address. The signals that latch each part of the address into the chip are RAS* and CAS*. Eight row address bits are placed on the input address (MA0 through MA7) and latched into the chip with RAS*. Next eight column address bits are placed on the input address (MA0 through MA7) and latched into the column address buffer by CAS*. In each case, the addresses on the input must be stable on or before the falling edge of active low RAS* and CAS*. The RAS* command is much like a chip enable because it affects both the row decoder and the sense amplifiers. CAS* acts like a chip select because it activates the column decoder and both input and output buffers.

Read Cycle

A read cycle begins with a valid row address on MA0 through MA7 and a negative-going transition of RAS* as shown by the timing diagram in Fig. 2-36. When RAS* reaches its active low condition, the row address is captured and latched into the row address buffer. Write enable (WE*) must be high to define a read operation. With WE* high, the data input MDn on pin 2 is disabled and the column address is placed on MA0-MA7. A few nanoseconds later (after the column address is stable, or valid), CAS* goes active low. This causes the column address to be captured and latched into the column address buffer. CAS* also turns on the input and output buffers so a data bit representing the logic value in the storage cell location identified by the intersection of the row and column address is copied onto the pin 14 data output (MDn). The data output remains valid until CAS* goes back high causing MDn to return to a high impedance condition.

The data out buffer has tristate capability. It provides output data of the same logic polarity as what was stored via the input. The output of

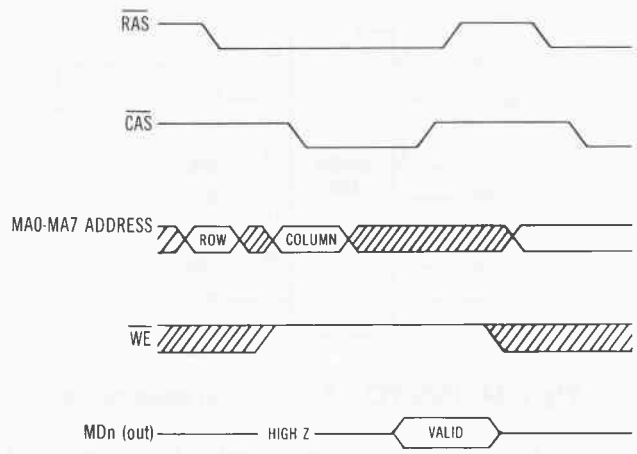


Fig. 2-36. Read cycle timing.

this buffer is a high impedance to the data bus until CAS* is pulled active low. During a read cycle, the output becomes active approximately 135 nanoseconds after the negative transition of CAS*. The output data then becomes valid. It remains valid as long as CAS* is active low. When CAS* goes high, MDn-pin 14 shifts to a high impedance state.

Write Cycle

Figure 2-37 shows the timing relationships for a RAM write operation. A write cycle begins like a read cycle in that a row address is placed on MA0-MA7, held stable, and then RAS* is brought low latching the row address into the row address buffer. The column address now appears on MA0-MA7, but here the difference occurs. WE* no longer remains (or goes) high signifying a read operation. Instead, WE* goes active low just before (or during) the time CAS* goes active low. If WE* goes low before CAS* becomes active low, the data out line (MDn on pin 14) holds in a high impedance state for the entire cycle.

The write cycle can occur in two ways: first, an "early write" occurs as described previously. In this operation, WE* must be valid long enough to be recognized by the read/write switching logic. The second write method is called a "delayed write" or "read-modify-write."

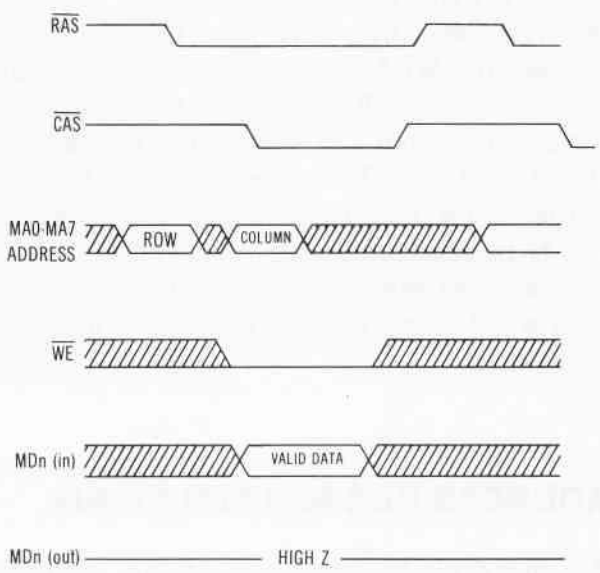


Fig. 2-37. Write cycle timing.

In this mode, WE* and MDn-in (pin 2) are delayed until after MDn-out (pin 14) becomes valid reading data out onto the data bus.

During an early write or delayed write operation, the falling edge of CAS* or WE* strobes the data in onto the data in buffer. During early write, WE* becomes active low before CAS* and the data is strobed into the chip by CAS*. During early write MDn-out (pin 14) is always tristate. In the delayed write or read-modify-write cycle, CAS* will already be low so the data is strobed into the chip by WE*. Here, the output follows the sequence for the read cycle.

Chip Select Circuitry

Figure 2-38 shows the circuitry associated with the row and column strobe signals that are used

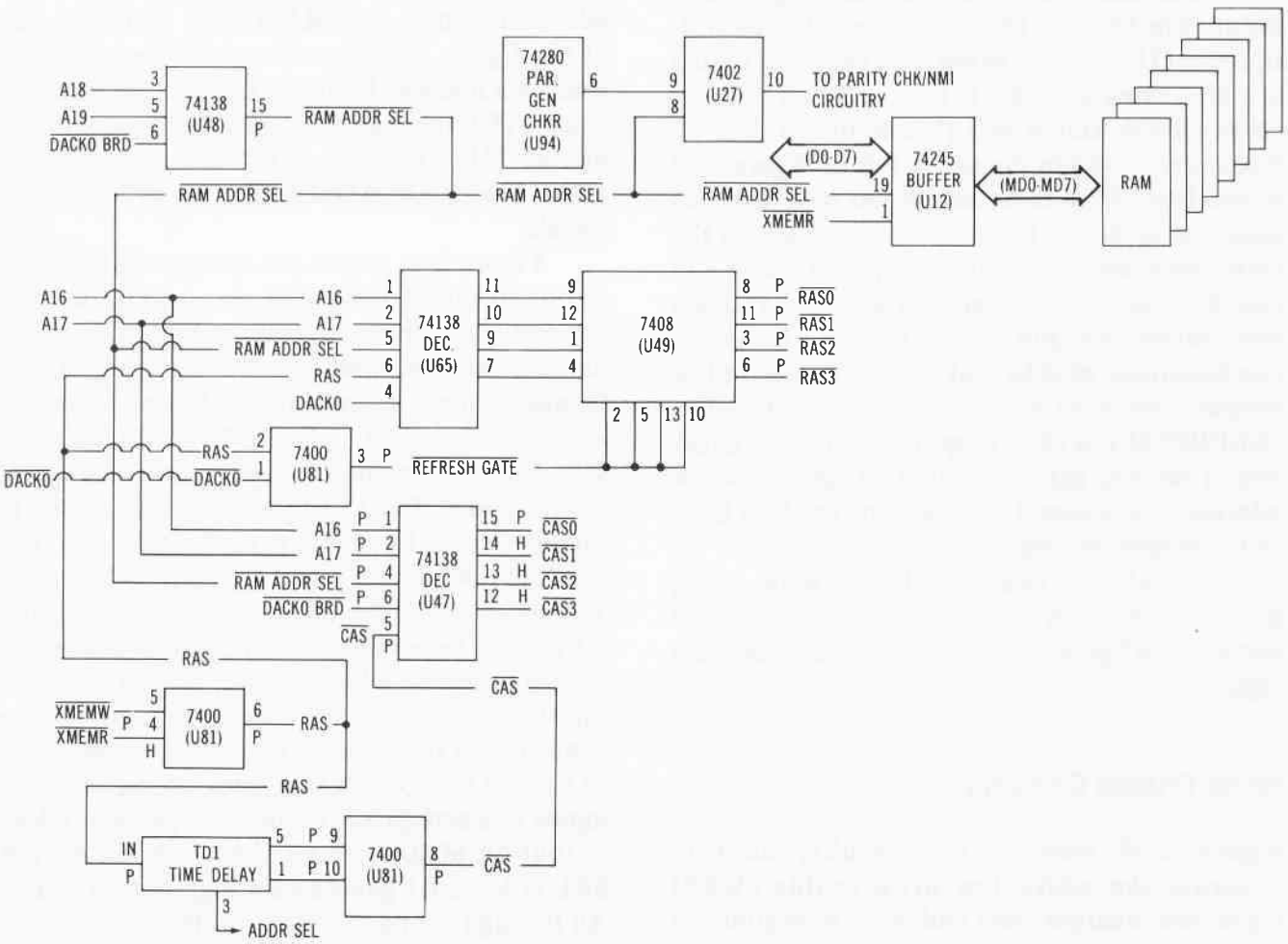


Fig. 2-38. RAM chip-select circuitry.

to select and operate the four banks of RAM memory. The occurrence of a buffered memory write (XMEMW*) or buffered memory read (XMEMR*) on pins 4 and 5 of 7400 quad 2-input NAND gate (U81) produces a high row address strobe signal (RAS) on output pin 6, as shown in the lower left of Figure 2-38. RAS is input to time delay TD1 where it becomes a delayed address select signal (ADDR SEL), and later, delayed inputs to pins 9 and 10 of U81 generating active low column address strobe (CAS*) on output pin 8. CAS* is used as an enable input to 74138 decoder (U47).

The RAS output from U81 is also connected to one input of U81 (pin 2) where it is NANDed with DACK0 on pin 1 to generate a (REFRESH GATE)* signal on pin 3. (REFRESH GATE)* becomes the qualifying input to each gate in 7408 quad 2-input AND gate (U49).

RAS also connects to the active high enable input (pin 6) of 74138 1-of-8 decoder/demultiplexer (U65). The other two enabling inputs are met when DACK0 is low, and ram address select (RAM ADDR SEL)* from the 74138 1-of-8 decoder (U48) in the upper left of Figure 2-38 is also low. With these conditions met, U65 will generate an active low output depending on the code comprised of inputs A16 (pin 1), and A17 (pin 2). Pin 3 (not shown) is tied high causing a low output on pin 7, 9, 10, or 11 for all combinations of A16 and A17. The active low output from U65 is ANDed with the active low (REFRESH GATE)* output from U81 to cause one of the AND gates to output an active low row address strobe signal (RASn*) for use by a bank of RAM memory chips.

Therefore, decoder U65 (with U49) produces the row address select signals, and decoder U47 produces the column address select signals.

Write Enable Circuitry

Figure 2-39 includes the circuitry used to generate the active low write enable (WE*) signal that enables read and write operations in the RAM memory. An active low buffered memory write signal (XMEMW*) is passed through 7404 hex inverter (U83) twice to produce the WE* signal that is applied to pin 3 of each dynamic RAM chip in the system and tristate enable pin 3 of 74125 quad 3-state buffer (U80). In U80, WE* is combined with the pin 5 output from 74280 parity generator checker (U94) to produce a parity bit (MDP) that is stored in the same row-column location as the data word being written into the other eight RAM chips.

ADDRESS BUS MULTIPLEXING

As shown in the lower left of Fig. 2-40, two 74LS158 quad 2-input data selector/multiplexers (U62 and U79) select row, and then column addresses for generating the MA0 through MA7 address inputs to each RAM chip. At the top left of the figure, 74245 octal transceiver (U12) is used as a one-way buffer to transfer the data bus signals (D0 through D7) onto the memory data bus as MD0 through MD7 under control of active low (RAM ADDR SEL)* and XMEMR* signals.

Figure 2-41 shows the internal architecture of one of the two multiplexers used to develop the row and column addresses. Multiplexer U62 handles the first nibble of these addresses. A0 through A3 are each connected to one input of internal AND gates 1, 3, 5, and 7. A8 through A11 are connected to one input of internal AND gates 2, 4, 6, and 8. Pin 15 is tied low enabling one of the two inputs to enable gates E1 and E2. The condition of address select (ADDR SEL) on pin 1 qualifies one of the other enable inputs to gates E1 or E2. Depending on which enable gate (E1 or E2) produces a high output, four of the eight input AND gates are selected passing the condition of the address bit at their other input to each of four output NOR gates. A zero on both inputs to a NOR gate is required to produce a high output on MA0 through MA3. When ADDR SEL is high, E1 generates a high output letting A0 through A3 pass through to the output (pins 4, 7, 9, and 12). The output signals MA0 through

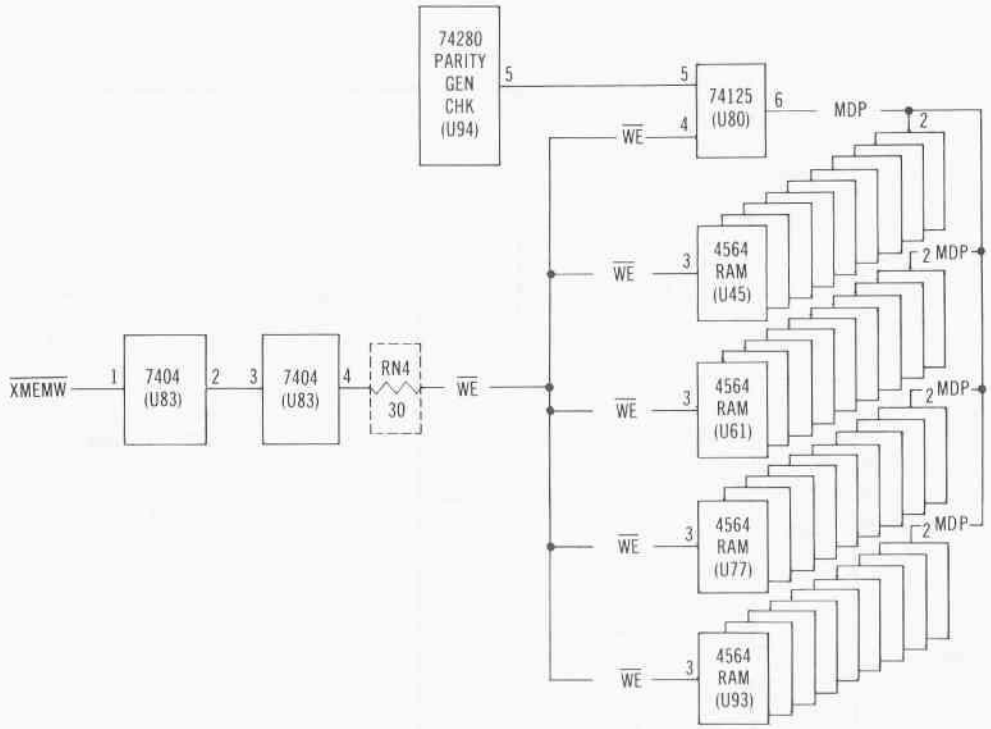


Fig. 2-39. Write enable (WE*) circuitry.

MA3 are actually the inverted condition of the qualified inputs.

The first address to pass through U62 and U79 is used to identify the row within each memory chip in a particular bank. Figure 2-42 shows that RAS is applied to time delay TD1 to produce an address select signal (ADDR SEL) that is passed to pins 1 of U62 and U79. The row addresses (A0 through A3) become valid (R valid) shortly after ADDR SEL goes high. When ADDR SEL returns low, the column address (A8 through A15) becomes valid (C valid) and is made available as MA0 through MA7 on the outputs of U62 and U79.

Table 2-9 shows the logic operation of U62. When (RAM ADDR SEL)* is high, no MEMW* signal will affect the output and no multiplexing occurs.

Figure 2-43 ties all the preceding control and enabling signals to describe the RAM addressing circuitry for bank 0. Row address select bank 0 (RAS0*) is generated by U49 and is held active low on the pin 4 input to each RAM chip in the array. The column address select bank 0 (CAS0*) signal is produced by U47. Multiplexers U62 and U79 produce the MA0 through MA7 address corresponding to the A0 through A7 row address input. When RAS0* occurs, the row address is latched inside each RAM chip. Then address select enable signal ADDR SEL goes low causing U62 and U79 to pass the A8 through A15 column address out as MA0 through MA7. CAS0* goes active low and the column address is latched into each RAM chip. Depending on the condition of write enable (WE*), data is written into each chip through pin 2 (MD0 through MD7), or read out of each chip via pin 14 (MD0 through MD7).

Table 2-9. Logic Table Relationship for 74LS158 Multiplexer U62

Enable Signal (RAM ADDR SEL)	Select Signal (MEMW)	Address Data (IN)	Address Data (OUT)
H	X	X X	H
L	L	A B	A
L	H	A B	B

A = input addresses A0-A3
B = input addresses A8-A11

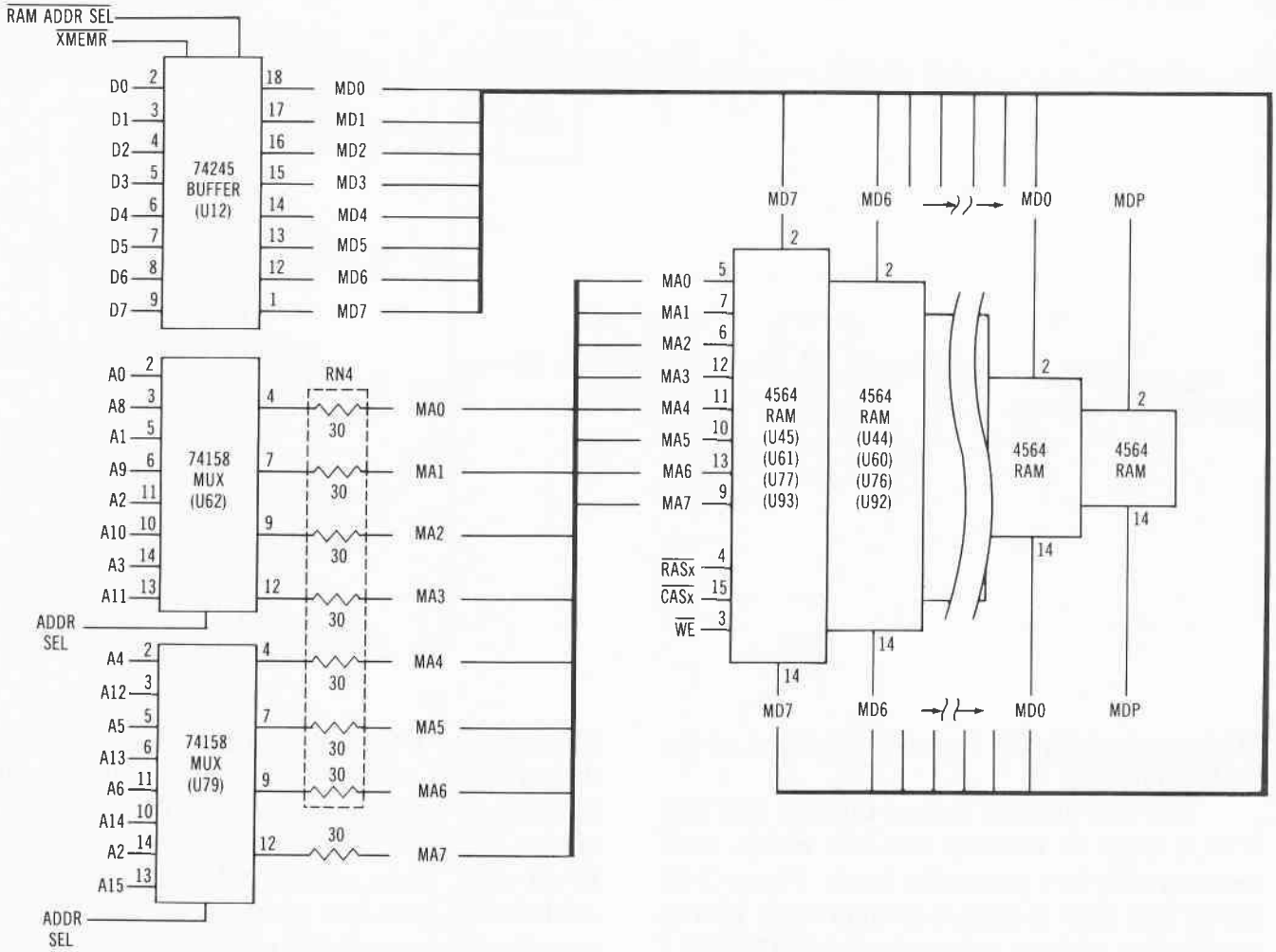


Fig. 2-40. RAM memory circuitry.

REFRESH

The RAM chips on your IBM PC system board are dynamic. Each memory cell is comprised of a single transistor that acts like a sample-and-hold circuit in which a capacitor is charged (logic 1) or discharged (logic 0). Since the charge in each cell can leak away, each memory location must be periodically accessed to restore the charge in cells that are in the logic high condition. The periodic accessing of memory to restore charge is called "refresh." The dynamic RAM (DRAM) chips in your system must be refreshed every 2-to-4 milliseconds.

Each time a DRAM location is read, all the memory cells in that row are refreshed. Capacitors that held a logic high are recharged to the full logic 1 charge. Depending on the manufacturer of the dynamic RAM in your system, refresh must occur each 2-to-4 milliseconds. Refresh is accomplished by doing a memory cycle at each of the 256 row addresses (8 bits = 256 address combinations) within each 2-to-4 millisecond interval. Any cycle in which RAS* becomes active will refresh the entire row. Because the chip output buffer is tristate to high impedance unless CAS* is applied, refresh can occur without ever activating CAS*. This RAS*-only refresh avoids any output during refresh and conserves power during the memory cycle. The RAS*-only refresh timing is shown in Fig. 2-44.

The counter OUT1 signal from the 8253 programmable interval timer (U34) shown in Fig. 2-45 clocks the 7474 dual-D latch (U67) to send a

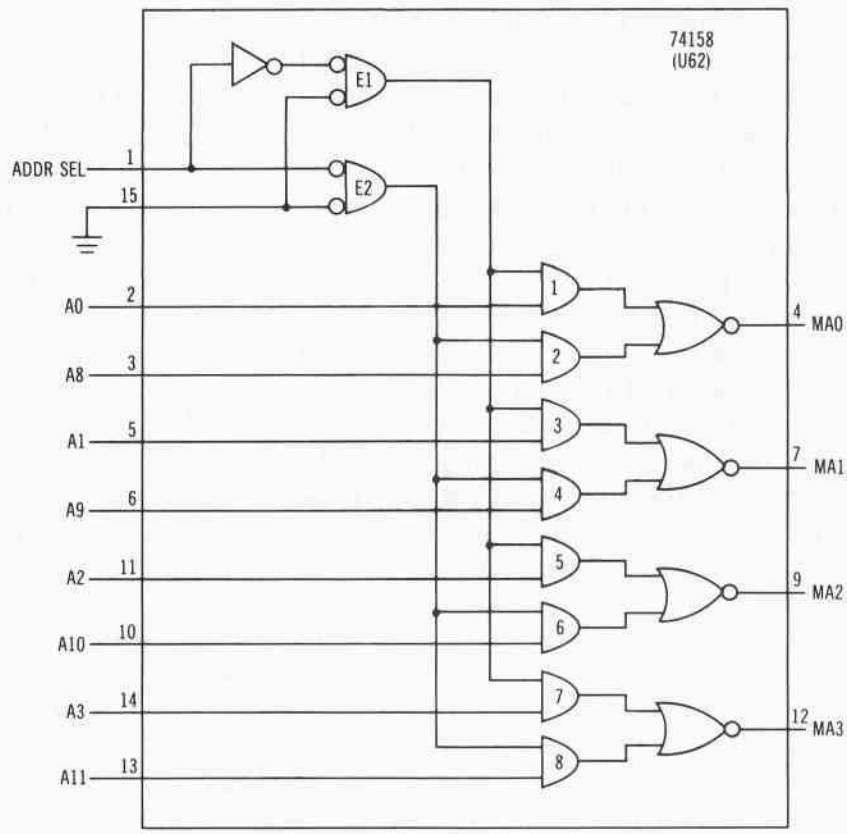


Fig. 2-41. The 74158 quad 2-input data selector/multiplexer.

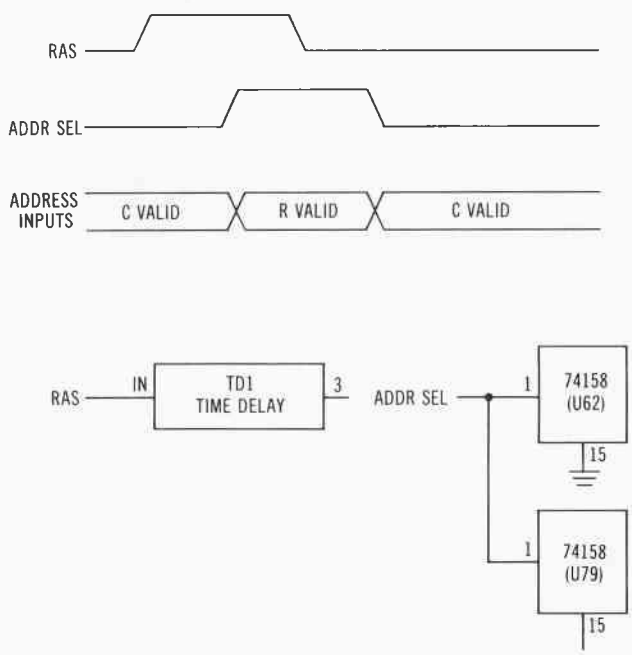


Fig. 2-42. Multiplex input selection timing and circuitry.

DMA request signal (DRQ0) into pin 19 of 8237 DMA controller (U35). The results in a (DACK0 BRD)* response out pin 25 that goes up to connect with pin 17 of 74244 buffer (U15) to become DACK0*. DACK0* combines with RAS in 7400 quad 2-input NAND gate (U81) to generate a (REFRESH GATE)* signal that enables the four gates in 7408 quad 2-input AND gate (U49) generating the row address strobe signals (RAS0*, RAS1*, RAS2*, and RAS3*) to the four dynamic RAM banks of memory. Each of these strobe signals represents the RASn* input to a particular RAM chip. Since OUT1 occurs every 15 microseconds, the memory accesses caused by OUT1 are well within the 2-to-4 millisecond refresh requirement. Refresh affects the memory on the system board and on any I/O expansion boards.

PARITY

Each bank of RAM chips includes one additional memory chip that is totally dedicated to storing a

bit corresponding to the parity of the 8 bits stored for each byte or word of data. In the IBM PC, each word of data written into memory is evaluated to determine the number of binary 1s in the word. If the number of 1s is an odd number, a 0 parity bit is stored in a special memory chip (parity memory). If the number of 1s is even, a 1 parity bit is stored in the corresponding location in the parity memory chip.

When 8 bits of data are read out of memory, the parity of the word is checked by special circuitry on the system board and compared to the parity bit stored in the corresponding location in the parity RAM. If the two parity values (computed and stored) are equal, the data word is accepted by the CPU or co-processor and system operation continues. However, if the two values differ, a non-maskable interrupt (NMI) occurs, causing the machine to halt with a "PARITY CHECK" display on the screen.

The key to parity generation and checking is the 74LS280 9-Bit odd/even parity generator/checker (U94). Figure 2-46 is a logic diagram of U94. The 74LS280 is used to detect errors in data retrieval from the RAM memory. Both even and odd parity outputs are available. If the number of data inputs that are high is even, the even parity output on pin 5 goes high. If the number of high data bits is odd, the odd parity output (pin 6) goes high, and the even parity output (pin 5) goes low.

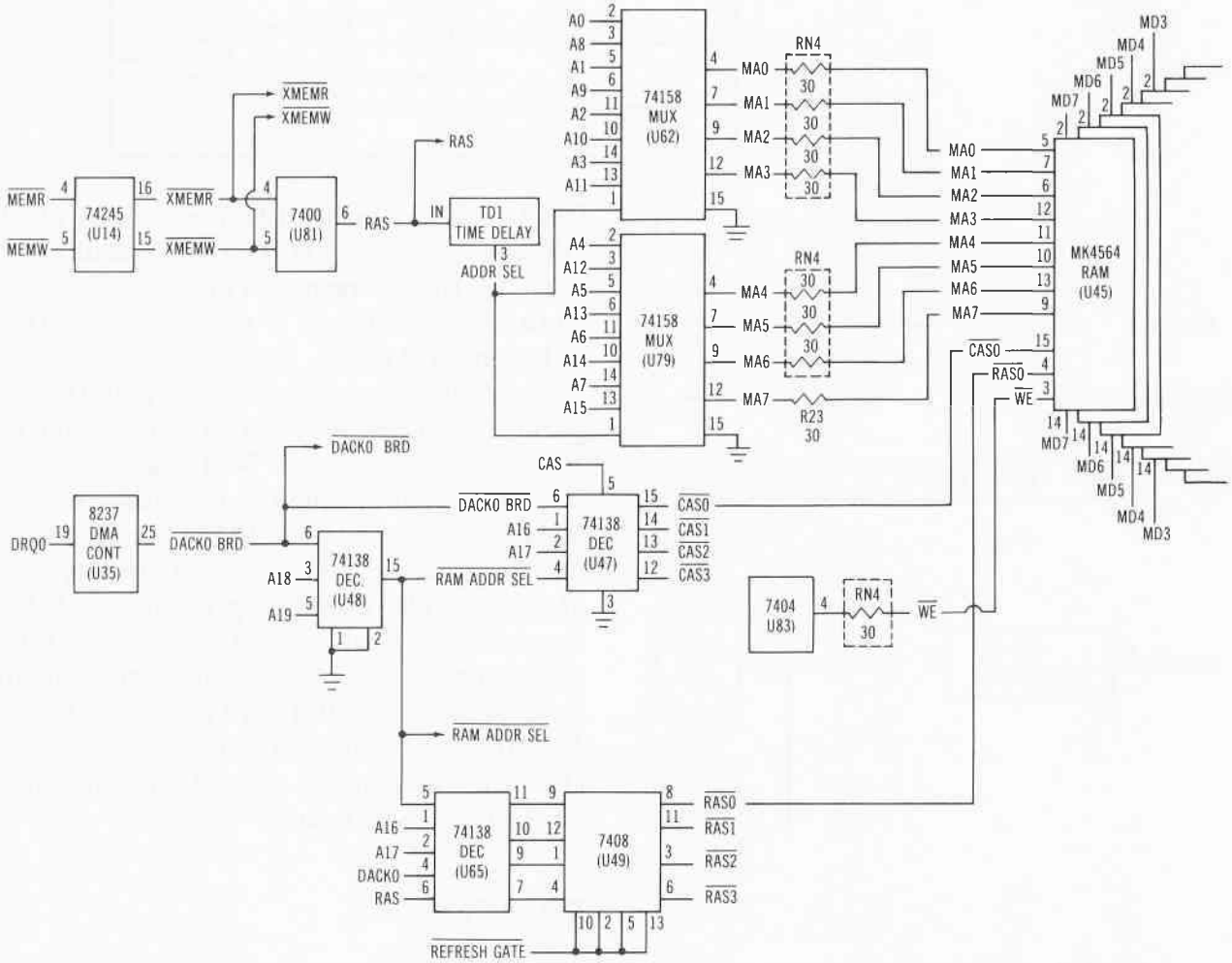


Fig. 2-43. RAM addressing circuitry–bank 0.

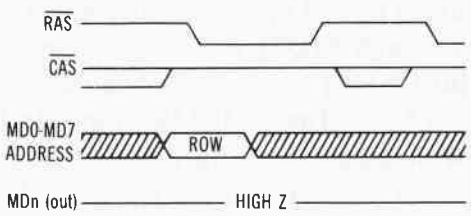


Fig. 2-44. RAS*-only refresh timing.

Both the even and odd parity outputs are used during memory writing and reading operations. The memory data bits (MD0 through MD7) that are being read into RAM are also applied to the inputs of U94 (pins 1, 2, and 8 through 13). The pin 4 input is generated by the ANDing of a parity bit read from memory and the status of the buffered memory read signal

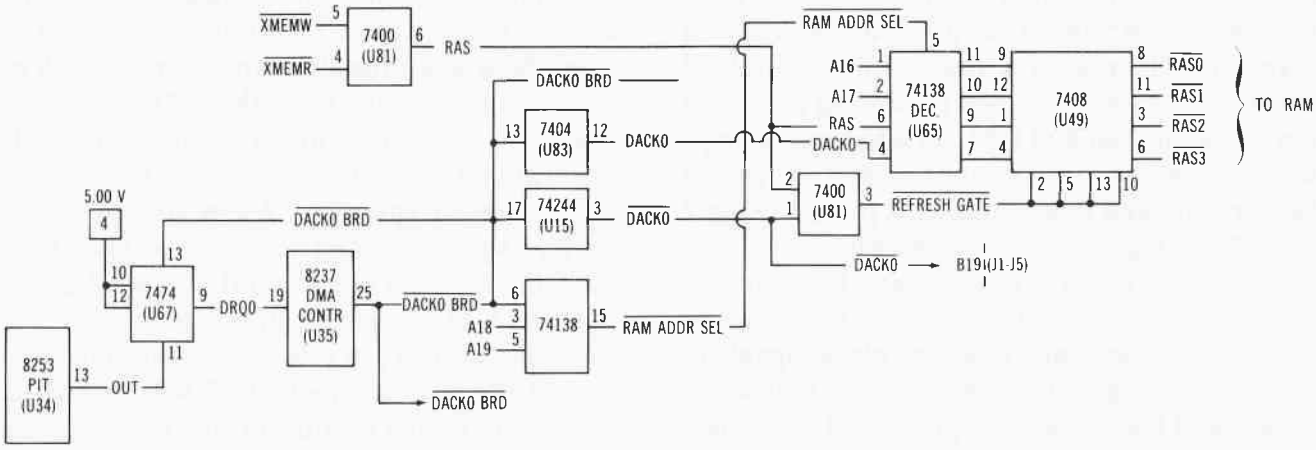


Fig. 2-45. Dynamic RAM refresh circuitry.

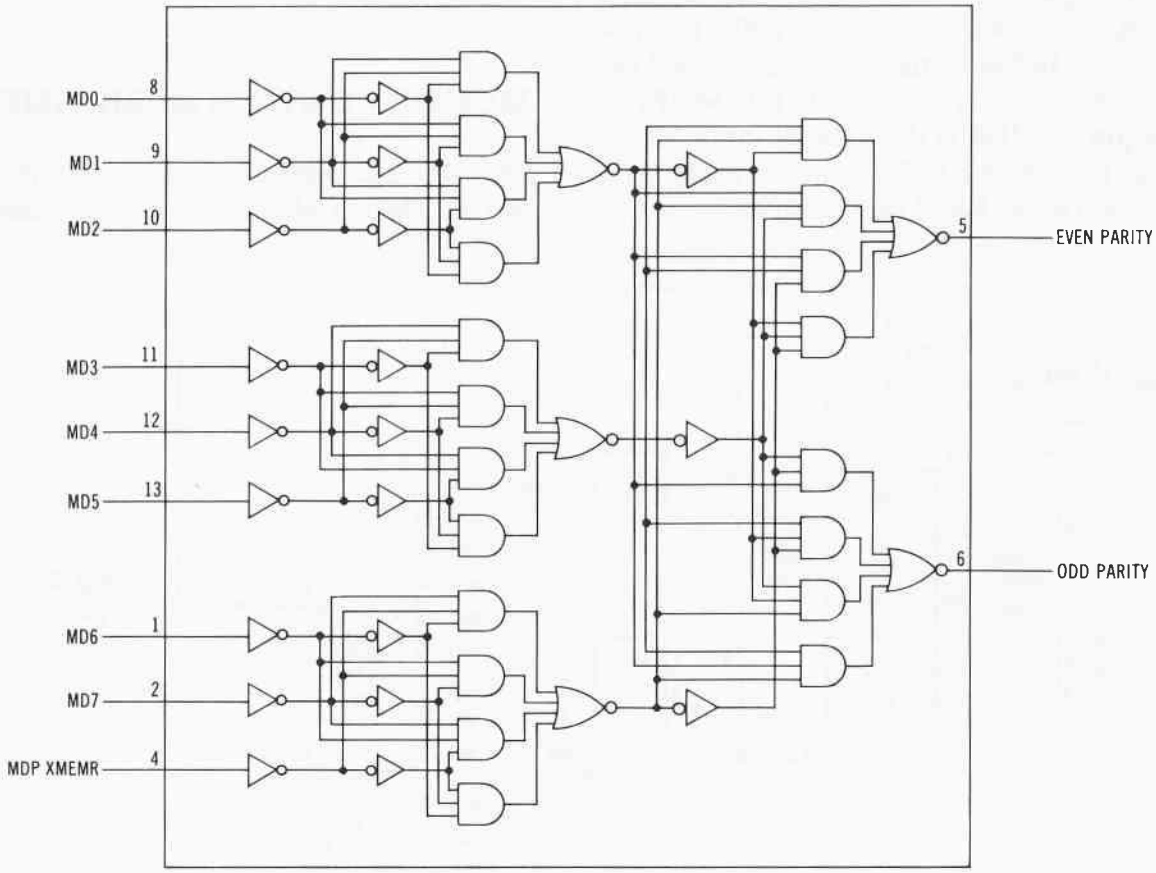


Fig. 2-46. Logic diagram of parity generator/checker U94.

XMEMR*. The nine inputs to U94 are passed through an inverter, buffer, AND, and NOR, circuit to yield outputs on pins 5 and 6 that correspond to the parity of the nine input data bits.

Figure 2-47 shows the parity circuitry in your PC. The buffered memory data bits MD6 and MD7 are combined with the pin 4 input from 74LS08 quad 2-input AND gate U97. This input is low during a write operation because the logic high XMEMR* input on pin 5 of 74LS04 hex inverter U83 places a low on pin 5 of U97 disabling the AND gate and producing a low out pin 6 and into pin 4 of U94. In addition, during a write operation, the output of each RAM chip is held tristate and this condition is passed to pin 4 of U97 as a high impedance open wire.

Parity generator/checker U94 is an asynchronous device that responds to signals on its inputs without any enabling clock signals to cause a timed operation. As such, the outputs on pins 5 and 6 are always represented. During a write operation, the pin 5 even parity output is passed into pin 5 of 74LS125 quad 3-state buffer U80. Active low write enable (WE*) is low enabling the buffer so the even parity signal can become the memory data parity bit (MDP) on output pin 6. MDP is the data bit input MD-in on input pin 2 of the MK4564 parity RAM (U37 in Fig. 2-47) in the RAM bank selected.

During a read operation, the 8 bits of data (MD0 through MD7) and the corresponding parity bit (MDP) are read out of memory and input into U94. The XMEMR* input to U83 is now low placing a high on pin 5 of U97. The combination of MDP and XMEMR produces an output signal from pin 6 of U97. This becomes one of the nine inputs to U94 during a parity checking operation. If the data word stored had an even number of 1s, the MDP value stored was high. Now when the data word is read back out of memory and into U94, the MDP high value causes the number of 1s to be odd. This generates a low out pin 5 and a high on odd parity output (pin 6). When this high (odd parity) signal is ORed with active low (RAM ADDR SEL)* on pin 8 of 74LS02 quad 2-input NOR gate U27, its pin 10 output is low.

However, if pin 6 is ever low during a read operation, both inputs to U27 will be enabled producing a high on output pin 10. This logic high will cause a non-maskable interrupt (NMI) to be generated.

MEMORY SWITCH ASSIGNMENT

Two DIP packages are mounted on the system board. Switch block 1 is used to identify the

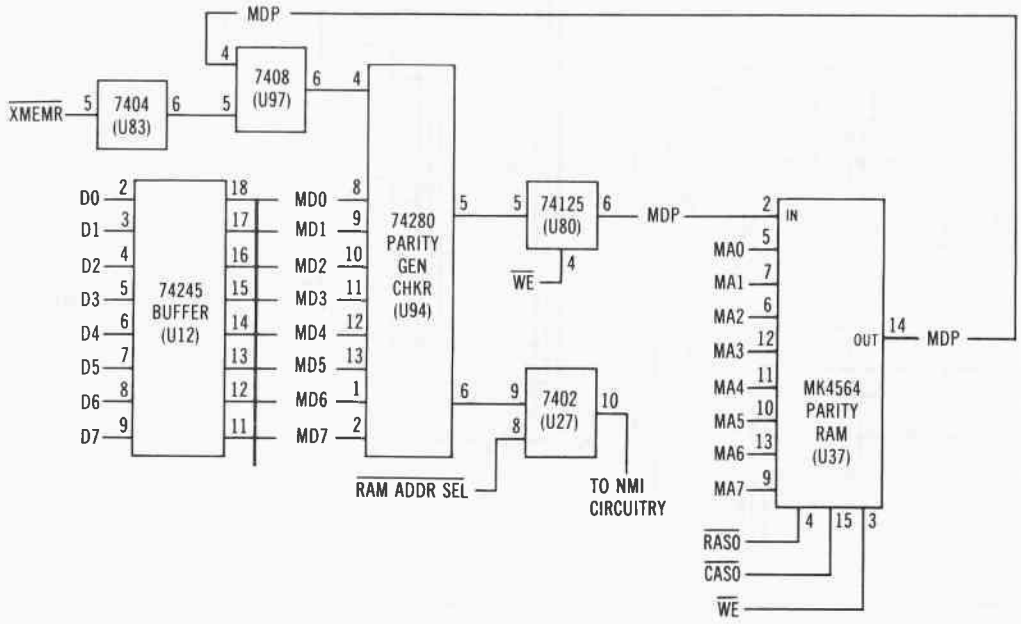


Fig. 2-47. RAM parity circuitry.

number of 5-1/4 inch diskette drives installed, the existence of an 8087 math coprocessor, the types of display connected, and the system board memory. Switch block 2 is used to define the amount of memory installed in the system.

Table 2-10 shows the function associated with each of the eight switches in blocks 1 and 2.

Table 2-10. System Board Switch Assignments

Switch	Function
	Switch Block 1:
1	Number of installed 5 1/4 inch diskette drives
2	Math coprocessor installed
3	System board memory
4	System board memory
5	Type of display connected
6	Type of display connected
7	Number of installed 5 1/4 inch diskette drives
8	Number of installed 5 1/4 inch diskette drives
	Switch Block 2:
1	Amount of installed memory
2	Amount of installed memory
3	Amount of installed memory
4	Amount of installed memory
5	Amount of installed memory
6	Always off
7	Always off
8	Always off

Figure 2-48 shows that both switches are connected to and read by the P8255A-5 programmable peripheral interface (U36). Each switch setting is manual. When each switch block is read by U36, a software program examines the input to U36 ports A and C and interprets the data to determine the type of system board configuration being accessed.

I/O MEMORY OPERATION

As described earlier, 64K of I/O memory space is available to the IBM PC system. Figure 2-49 shows the I/O device decoding circuitry. Two primary chips are involved in generating the chip select/decode signals. On the far left, 74LS245 octal transceiver (U14) has active low IOR* and IOW* on its input pins 2 and 3. Its chip enable pin 19 is tied to ground constantly qualifying U14 for operation. Send/receive control pin 1 has inactive high (DMA AEN)* connected. With pin 1 high and pin 19 low, U14 is configured to pass data from left to right. The output signals from U14 are labelled XIOR* (pin 18), and XIOW* (pin 17). Active low buffered I/O read (XIOR*) and active low buffered I/O write (XIOW*) act as qualifying signals for various chips in Fig. 2-49. In the center of Fig. 2-49, XIOR* is connected to input pin 12 of 74LS02 quad 2-input NOR gate (U27). This signal combined with XA9, and (ROM ADDR SEL) ANDed with XMEMW determine the logic value of direction (DIR) out pin 1 of U27. When address enable board (AEN BRD) is low, the value of DIR determines the direction of information flow through 74LS245 data bus octal transceiver (U13). When XA9 and the other three signals that generate DIR are low, DIR becomes low causing U13 to read data from left (XD0 through XD7) to right (D0 through D7) into the 8088 CPU (U3).

Besides U14, the other critical chip in the Fig. 2-49 select and decode circuitry is 74LS138 3-of-8 decoder U66. Three inputs are applied to U66: XA5 (pin 1), XA6 (pin 2), and XA7 (pin 3). The signals that enable U66 are XA8 (pin 5), XA9 (pin 4), and AEN* (pin 6). To enable U66, XA8 and XA9 must both be low–AEN* must be high. When these conditions are met, the combination of XA5, XA6, and XA7 will cause one of the eight (six of eight used) outputs to go active low.

The 8088 CPU (U3) can address up to 64K I/O ports using the lower 16 bits of its address bus (AD0 through AD7 and BA8 through BA15). Notice that the IBM PC only uses XA5 through XA9 for I/O port addressing. XA0 through XA4, and the buffered addresses above XA9 are ignored. This results in redundant addressing in that 32 different combinations of XA0 through XA9 can cause each of the U66 outputs to go active low. If XA5 through XA9 are all low, and AEN* is high, pin 15 is pulled low producing an active DMA chip select (DMA CS)* signal for the 8237 DMA controller (U35). However, with

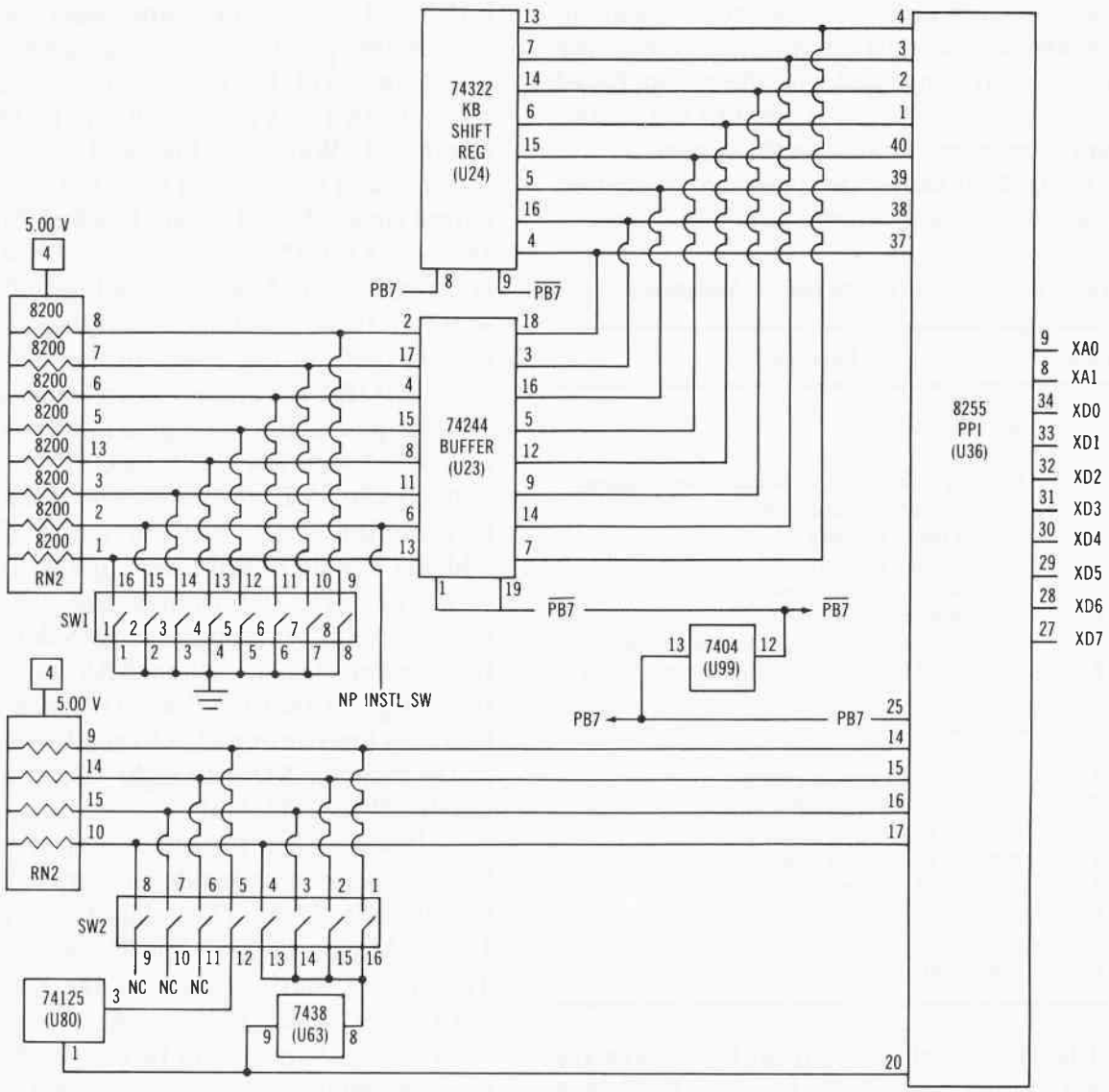


Fig. 2-48. Switch block read circuitry.

these same input conditions, XA0 through XA4 can cycle through 32 different combinations and still produce the same pin 15 (DMA CS)* output.

If U66 is enabled, and the XA5 through XA7 input is high-low-low, pin 14 goes active low producing interrupt chip select (INTR CS)* for the 8259 programmable interrupt controller (U2). Similarly, a low-high-low input combination will produce an active low terminal/count chip select (T/C CS)* for the 8253 programmable interval timer (U34). Another combination of XA5 through XA7 produces the active low pin 12 PPI chip select (PPI CS)* signal for the programmable peripheral interface (U36).

A low-low-high input produces an active low output on pin 11. This output is passed to pin 12 of 74LS02 quad 2-input NOR (U50). If XIOW* is also low (an output instruction is being executed), pin 13 of U50 goes high. This signal is inverted by 74LS04 (U51) to produce an active low (WRT DMA PG REG)* input to pin 12 of the 74LS670 DMA page register (U19).

A high-low-high input to U66 produces an active low output on pin 10. This signal is passed to pin 8 of another NOR gate in 74LS02 (U50). Its companion pin 9 input is XIOW*. If XIOW* is also active low, (WRT NMI REG)* goes active low on the clock input (pin 11) to 74LS74 dual

D-latch (U96). When (WRT NMI REG)* returns high, the positive transition of this signal transfers the value of XD7 to its Q output in the non-maskable interrupt (NMI) circuitry. This makes the NMI software maskable.

INTERRUPTS

The sequential execution of instructions by 8088 CPU U3 can also be altered by special input signals (NMI and INT) on pins 17 or 18. CPU interrupts can be generated by the hardware or initiated by software during program execution. Hardware interrupts are classified as maskable (can be ignored by design) or non-maskable.

The recognition of an interrupt by 8088 CPU U3 causes the machine to transfer control to a new program location. The 1024 memory locations (00H through 3FFH) are dedicated to storing up to 256 four-byte interrupt vectors which contain address pointers to the interrupt routines stored elsewhere in the memory space of the system. Each vector is comprised of a 16-bit base address in the higher-addressed word for the code segment register and a 16-bit offset in the lower-addressed word for the instruction pointer register inside the CPU, as shown in Fig. 2-50. Taken together these two registers point to a particular vector memory location where an interrupt service routine is stored. Each vector is 4 bytes long so the CPU can calculate the starting location of the correct interrupt code entry by multiplying the type code by four (for example, type 1 x 4 equals location 004H).

Four classes of interrupts are possible in the IBM PC system:

1. CPU internal interrupt
2. Non-maskable interrupt
3. External interrupt
4. Single-step interrupt

The four interrupt sources for U3 are shown in Fig. 2-51.

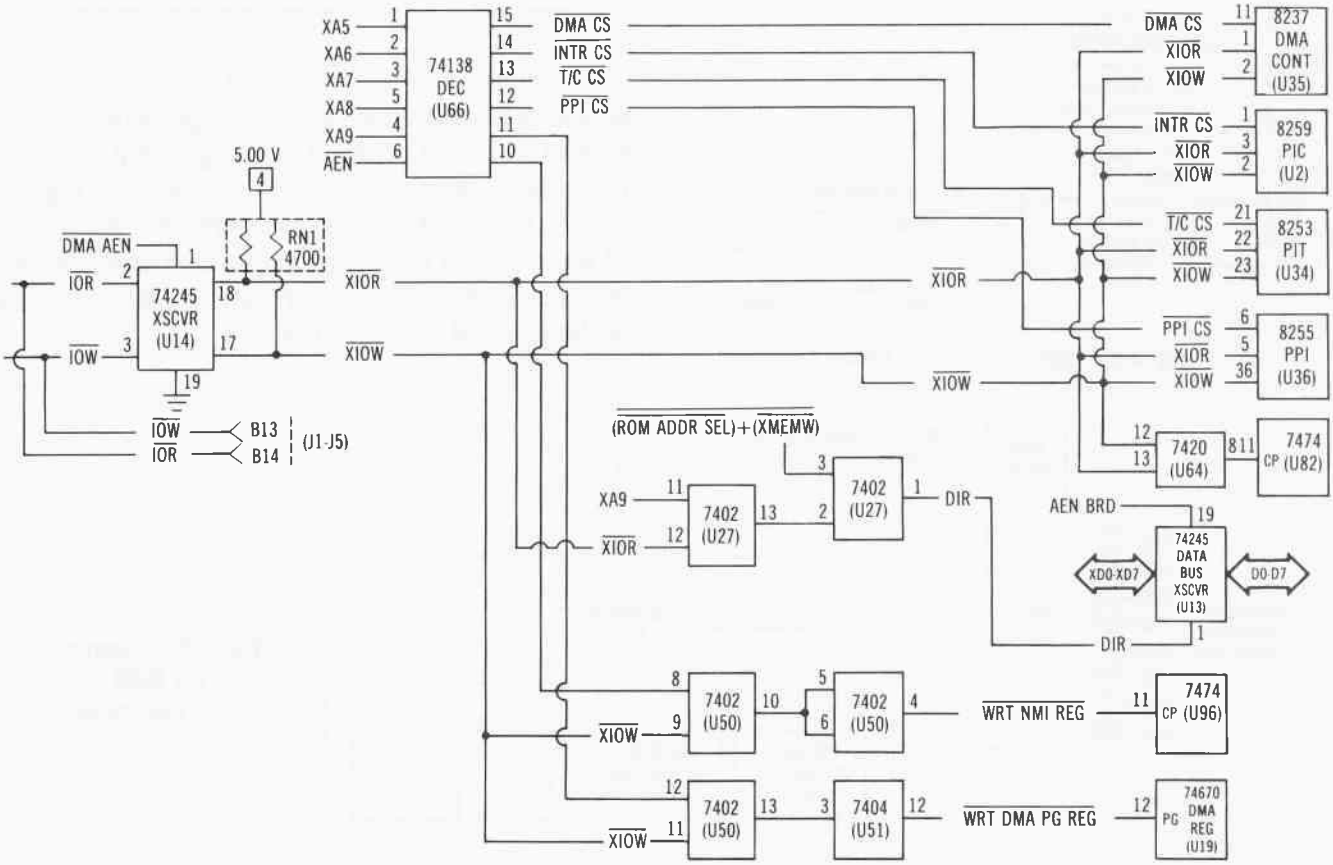


Fig. 2-49. I/O read/write chip select and decoding circuitry.

Interrupts have an internal priority for servicing by U3. Internal interrupts such as divide error, program interrupt, and overflow have the highest priority followed by the non-maskable, external device, and internally programmed single-step interrupts.

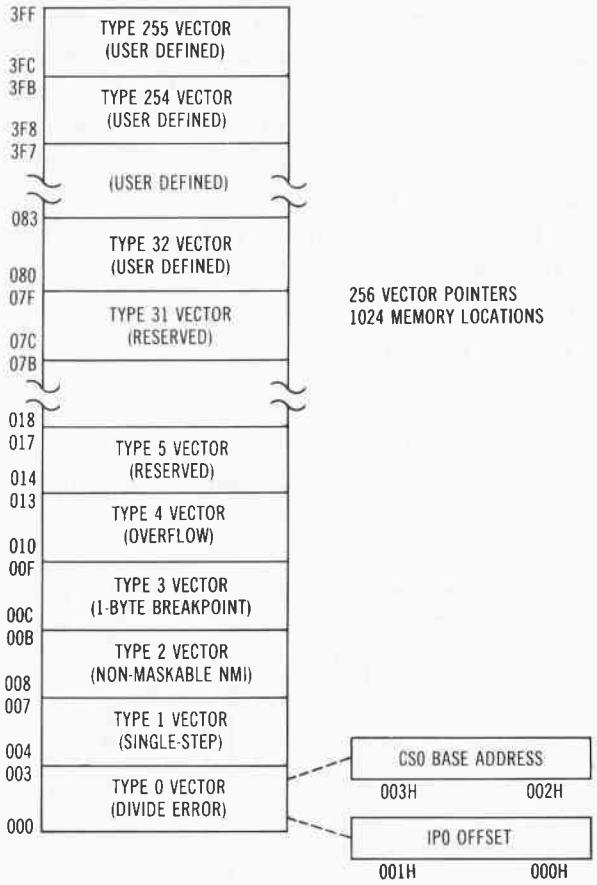


Fig. 2-50. Interrupt vector table.

When an interrupt occurs, U3 completes its current instruction and then begins to service the interrupt(s) on a priority basis. The interrupt latency (interval between CPU recognition of interrupt request and execution of initial interrupt routine) including time to save CPU status and register contents (interrupts only save CS, IP, and flag registers automatically) can vary between 50 and 61 clock cycles as shown in Table 2-11.

Table 2-11. Interrupt Priority and Processing Times

Interrupt	Priority	Description	Internal/ External	Minimum Servicing Time (clocks)
Type 0	1	Divide error	Internal	51
Type 1	4	Single step	Internal	50
Type 2	2	Non-maskable	External	50
Type 3	1	1-byte breakpt	Internal	52
Type 4	1	Overflow (into)	Internal	53
Software	1	Program (intn)	Internal	51
Hardware	3	Maskable (int)	External	61

Each time an interrupt occurs, certain action takes place within U3 depending on the type of interrupt. Except for single step, internal interrupts cannot be disabled. With a higher service priority for internal interrupts, the arrival of an external interrupt request at the time an internal interrupt occurs results in the servicing of the internal interrupt first.

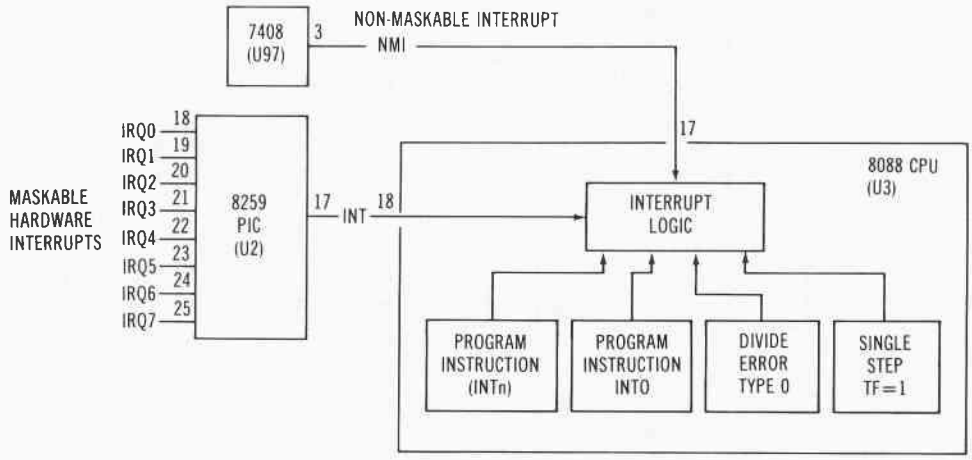


Fig. 2-51. Sources for 8088 CPU interrupts.

Internal Interrupts

Internal interrupts can be generated by program instructions INT and INTO, by error conditions resulting from DIV or IDIV instructions, and by single stepping each instruction when the trap flag (TF) has been set in the control flag portion of the 8088 execution unit. Setting TF to high causes the CPU to cause a type 1 interrupt after execution of each instruction. This single-step operation for most instructions is useful for program debugging.

A programmed interrupt (INTn instruction) and other internal interrupts cause the interrupt code contained in the instruction or a predefined interrupt code to be recognized by the CPU.

External Interrupts

Hardware interrupts from external devices are achieved in two ways. Parity errors result in the generation of a non-maskable interrupt signal (NMI) that is applied to pin 17 of the 8088 CPU U3. All other hardware interrupts are generated via interrupt request input signals to a 8259 interrupt controller (U2), as shown in Fig. 2-52. Chip pin descriptions appear within the 8259 block. IBM PC chip pin assignments are on each line connected to the block from outside. The output of U2 is a CPU interrupt INT applied to pin 18 of 8088 U3.

8259 Interrupt Circuitry

The 8259 programmable interrupt controller is a 28-pin DIP (dual in-line package) that handles up to eight vectored priority interrupt requests and generates a logic high interrupt signal to the CPU. The asynchronous logic requires no clock input. It responds to any interrupt request input given the control lines are properly configured. Its ability to handle multilevel priority interrupts minimizes software and real-time system overhead.

A description of the function of each active pin on U2 is provided in Table 2-13. Some pins (12, 13, and 15) are not used in this configuration.

Table 2-12 lists the IBM PC interrupt vector assignments.

Table 2-12. IBM PC Interrupt Vector Assignments

Interrupt Type	Function
0	Divide by zero error
1	Single-step
2	Non-maskable interrupt (NMI)
3	One-byte breakpoint
4	Overflow
5	Invokes logic to print screen
6	Reserved
7	Reserved
8	Time of day (18.2/second)
9	Keyboard hardware interrupt
A	Reserved
B	Serial communications
C	Serial communications
D	Fixed disk hardware interrupt
E	Diskette hardware interrupt
F	Printer hardware interrupt
10	Video input/output call
11	Equipment check call
12	Memory check call
13	Diskette input/output call
14	RS-232 input/output call
15	Cassette input/output call
16	Keyboard input/output call
17	Printer input/output call
18	ROM basic entry code
19	Boot strap loader
1A	Time of day call
1B	Get control on keyboard break
1C	Timer interrupt control
1D	Video initialization table pointer
1E	Diskette parameter table pointer
1F	Graphics character generator pointer
20	DOS program terminate
21	DOS function call
22	DOS terminate address
23	DOS CTRL-BRK exit address
24	DOS fatal error vector
25	DOS absolute disk read
26	DOS absolute disk write
27	DOS terminate, fix in storage
28-3F	DOS (reserved)
40-5F	Reserved
60-67	User software interrupts
68-7F	Not used
80-85	BASIC interrupts (reserved)
86-F0	Used by running BASIC interpreter
F1-FF	Not used

Table 2-13. 8259 Programmable Interrupt Controller (U2) Pin Assignments

Symbol	Pin No.	Type	Name and Function
(INTR CS)*	1	I	Interrupt chip select: Active low signal that enables XIOR* and XIOW* communication between the 8088 CPU and the 8259 PIT.
(XIOW)*	2	I	External I/O Write: Active low when (INTR CS)* is low enables PIT U2 to receive command words from CPU U3.
(XIOR)*	3	I	External I/O Read: Active low when (INTR CS)* is low enables PIT U2 to place CPU status data on address/data bus. Bidirectional Data
AD0-AD7	4-11	I/O	Bus: Transfer path for control, status, and interrupt vectors.
GND	14		Ground:
(SP/EN)*	16	O	Slave program/enable buffer: Used in buffered mode as output to 7410 three-input NAND U84 which combines this signal with DEN from the 8288 bus controller to generate G*, the enable signal for 74245 transceiver U8.
INT	17	O	Interrupt: Active high when valid interrupt request is asserted. Connected to CPU interrupt input pin 18.
IRQ0-IRQ7	18-25	I	Interrupt request: Active high asynchronous inputs used to generate interrupt output (INT) to CPU.
INTA*	26		Interrupt acknowledge: Response from CPU that causes 8259 to enable interrupt vector data onto address/data bus.
XA0	27	I	Buffered address bit 0: Acts with (INTR CS)*, (XIOR)* and (XIOW)* to decode CPU U3 command words and provide status back to U3.
VCC	28	I	Supply voltage: +5.00 volts

During power-up, the BIOS software does a test on U2. It writes all zeroes into an interrupt mask register inside the 8259 and compares the value read with what was written. Then it disables device interrupts and writes all ones to this register. Again, the program reads the register contents and compares it with what was written. All 1s in this register cause all interrupt request lines to be masked. The program reenables external interrupts and waits for 1 second to check for any hot interrupts that may have occurred during the test. If an error occurs during the comparisons, the speaker beeps and the program halts the system.

When several devices attempt to interrupt the system operation at the same time, a priority has been established at the input request lines to U2. The device with the highest priority is serviced first. Higher priority inputs can be masked without affecting the interrupt request lines of lower priority inputs. IRQ0 on pin 18 is the highest priority interrupt. It is used to cause a time of day clock tick. The second highest priority interrupt (IRQ1) is used during keyboard entry. Table 2-14 lists the assignments for all eight interrupt request inputs to U2. If two request lines become active at the same time, the request whose number is closest to IRQ0 is serviced first.

Table 2-14. 8259 Hardware Interrupt Request Listing

Interrupt Request	Type of Interrupt	Function
IRQ0	Type 8	Timer
IRQ1	Type 9	Keyboard
IRQ2	Type A	Reserved
IRQ3	Type B	- Asynchronous communications (secondary) - Synchronous data link control (SDLC) communication - Binary synchronous communications (BSC)
IRQ4	Type C	- Asynchronous communication (primary) - SDLC communication - BSC (primary)
IRQ5	Type D	Fixed disk
IRQ6	Type E	Diskette
IRQ7	Type F	Printer

Figure 2-53 shows the external interrupt circuitry associated with 8259 U2. During initiali-

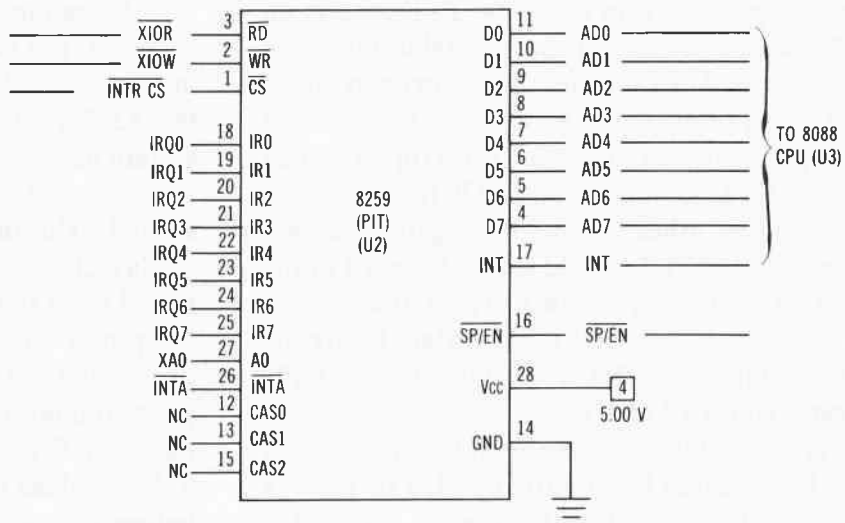


Fig. 2-52. 8259 programmable interrupt controller (U2).

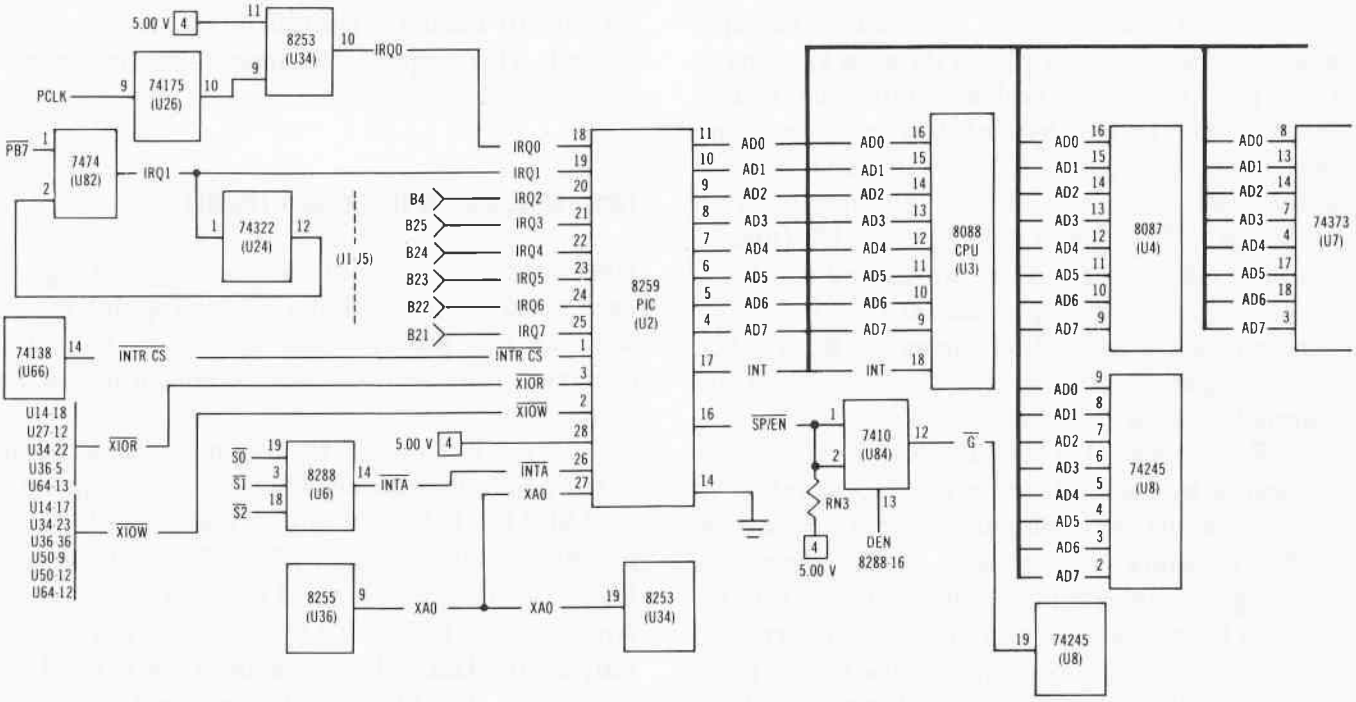


Fig. 2-53. 8259 programmable interrupt controller (U2) external hardware interrupt circuitry.

zation, U2 is configured so interrupt-request signals must make a low-to-high transition to generate INT out pin 17. The chip is also set to recognize that it is the only 8259 in the system. The chip is set to buffered mode causing (SP/EN)* to be an output from pin 16.

Interrupt Sequence

Upon sensing an active high interrupt-request signal on one of its eight inputs, U2 sets a corresponding bit in an interrupt-request register latching each interrupt request. U2 then passes an interrupt signal (INT) out pin 17 to pin 18 of CPU U3 causing the 8088 to pull status outputs S0*, S1*, and S2* active low (pins 19, 3, and 18 respectively). This code is passed to 8288 bus controller (U6) causing it to generate an active low interrupt acknowledge (INTA*) signal on pin 14.

This first INTA* signal on pin 26 of U2 communicates to the 8259 that the request has been honored by U3, and that an interrupt-

acknowledge cycle is in progress. INTA* valid on pin 26 causes the 8259 programmable interrupt controller (U2) to set the highest priority bit in an interrupt-service register and to clear its corresponding bit in the interrupt-request register. CPU U3 also brings LOCK* low on pin 29 preventing other devices from gaining access to the bus. LOCK* is held active from T2 of the first INTA* bus cycle until T2 of the second INTA* bus cycle. CPU U3 also floats its address/data bus. INTA* is active during CPU bus cycle states T2 and T4.

A second 000 status signal is generated by CPU U3 causing bus controller U6 to place a second INTA* signal on U2 pin 26. When the 8259 (U2) senses this second interrupt acknowledge signal, it places data on U2 pins 4 through 11 of the address/data bus (AD0 through AD7). This byte of data represents the interrupt type (type 0 through 255) that is associated with the device requesting the interrupt. This byte is read by CPU U3 (pins 9 through 16) where it is multiplied by four producing an address pointer into the interrupt vector table. Figure 2-54 shows the 8259 (U2) edge-triggered timing relationships during external hardware interrupt activities.

Before 8088 U3 calls the interrupt routine indicated by the address in the vector table, it saves its status by pushing the current contents of its flag registers onto its stack. It then clears the flag register interrupt enable and trap bits to prevent later maskable and single step interrupts, and establishes an interrupt routine return path by pushing the current CS and IP register values onto the stack before loading these registers from the vector table.

Once the highest priority interrupt has been serviced, the next highest priority interrupt (as indicated by the next highest priority bit set in the 8259's interrupt-request register) receives system attention. One interrupt request (IRQ0) is produced approximately 18.2 times each second. This interrupt is used to produce a time-of-day clock tick. The 2.386363 MHz PCLK signal from the clock generator (U11) is sensed on pin 9 of 74LS175 quad-D flip-flop U26 producing a 1.19318 MHz clock input to 8253 programmable interval timer (U34) pin 9. Counter 0 inside U34 is preset so it accepts 65536 pulses of the 1.19318 MHz clock input before it produces an output. Counter 0 output on pin 10 occurs 1193180/65536 times each second. This output is the time-of-day interrupt-request signal (IRQ0).

Non-Maskable Interrupt (NMI)

NMI is an asynchronous edge-triggered signal used to tell the CPU that a "catastrophic" event such as bus parity error detection has just occurred. The NMI circuitry is shown in Fig. 2-55.

A parity output from pin 6 of 74LS280 parity generator checker U94 is matched up with (RAM ADDR SEL)* in a 74LS02 NOR U27 to produce a latch input to 74LS74 dual-d latch U96. (XMEMR)* is used to clock U96 pin 3. An inverted (ENB RAM PCK)* from pin 6 output of 74LS04 hex inverter U99 is used to clear U96. The Q* output from pin 6 of U96 is passed to input pin 10 of 74LS10 NAND U84. A second input to U84 comes from pin 1 of a

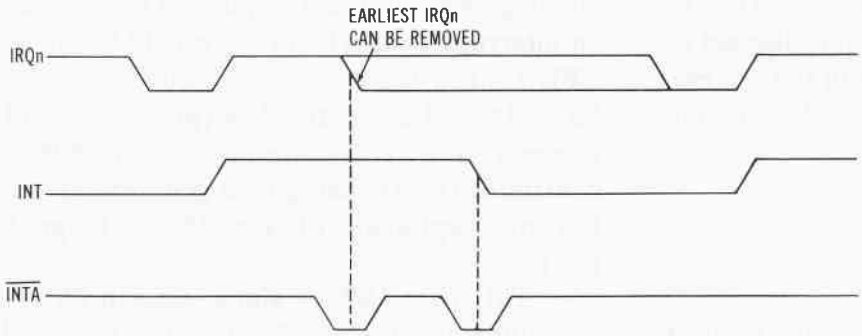


Fig. 2-54. 8259 interrupt timing relationships.

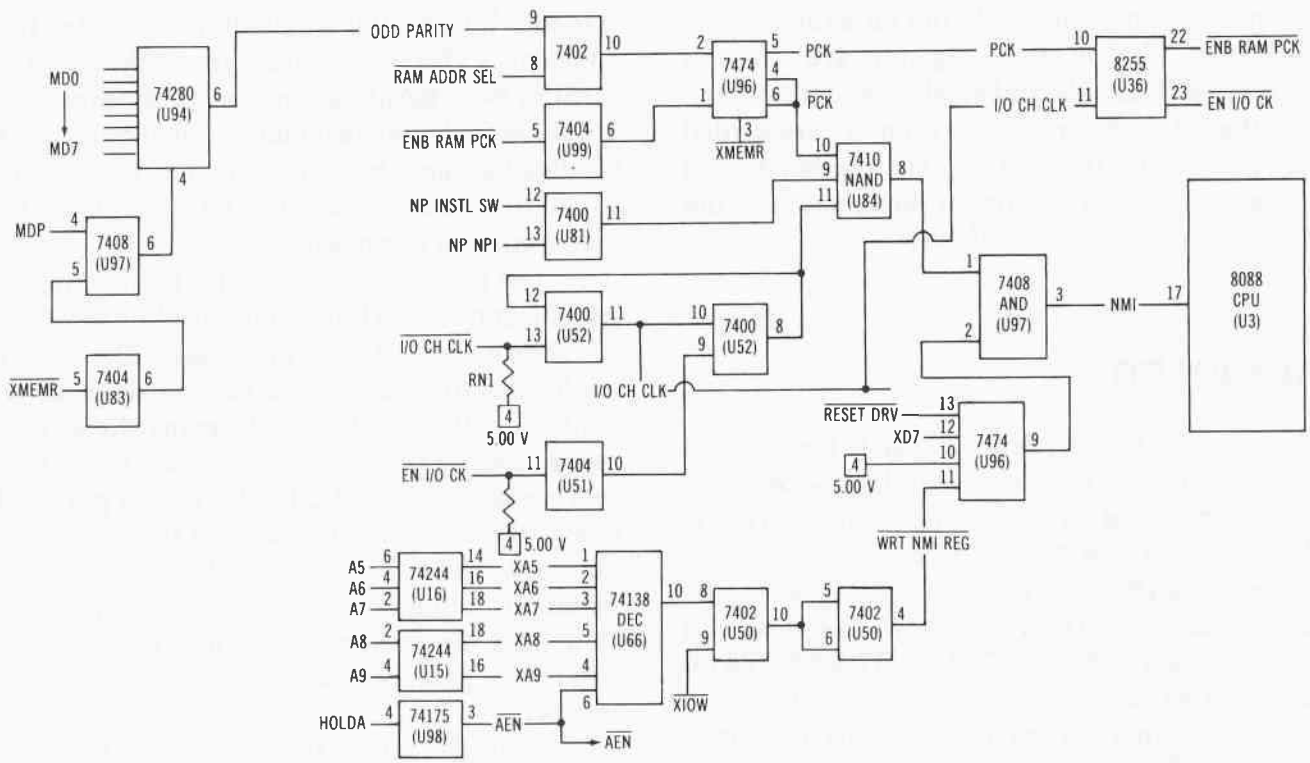


Fig. 2-55. Non-maskable interrupt (NMI) circuitry.

74LS00 2-input NAND U81 which combines NP INST SW and NP NPI on pins 12 and 13 to produce an enabling signal on pin 9 of U84. Pin 11 of U84 is enabled by output pin 8 from 74LS00 2-input NAND U52. U52 was enabled by I/O CH CLK and EN I/O CK.

The output from 74LS10 U84 pin 8 becomes one of two inputs to 74S08 AND U97. The other input (pin 2) to U97 comes from an address decoding circuitry comprised of 74LS244 tristate buffers U15 and U16, 74LS175 quad-D latch U98, 74138 Decoder U66, 7402 2-input NOR U50, and 74LS74 dual-D latch U96.

The presence of logic 10100 in address bits A5, A6, A7, A8, and A9 respectively on the inputs to the 74LS244 tristate buffers U15 and U16, with the AEN/(AEN BRD)* flip-flop in quad-D flip-flop 74LS175 reset holding (AEN)* high causes the "five-not" (5*) 74138 decoder output on pin 10 to go low enabling the other half of 74LS02 NOR (U50). This input on pin 8, and (XIOW)* active low on pin 9 causes output pin 4 to generate an active low (WRT NMI REG)* input to clock 74LS74 dual-D flip-flop U96. XD7 is latched into U96 producing an output on pin 9 that is passed to input pin 2 of 74S08 AND U97. Once enabled, 74S08 U97 output on pin 3 is a non-maskable type 2 interrupt (NMI) signal that is passed to input pin 17 of the 8088 CPU U3.

NMI is the highest priority hardware interrupt. NMI cannot be masked or disabled as can the hardware interrupts from 8259 PIT U2. CPU U3 input line 17 is edge-triggered. Therefore, to prevent spurious positive transitions on pin 17 from appearing like an NMI, this signal is held for two clock cycles to guarantee recognition by U3.

The asynchronous occurrence of NMI on pin 17 causes CPU control to transfer to an interrupt service routine defined by type 2 vector following the completion of the CPU's current instruction execution. NMI is predefined as a type 2 interrupt; therefore, the CPU is not supplied with a type code to call the NMI service procedure. U3 also does not produce INTA bus cycles in response to this signal.

During the CPU's internal acknowledgment of NMI, the current contents of the flag register

are pushed onto the stack, the interrupt enable and trap bits in this register are cleared (disabling maskable and single-step interrupts), and the NMI vector address pointers are loaded into the CS and IP registers. The combination of these two registers points to the location of the NMI interrupt service routine.

KEYBOARD

Two types of keyboards are available for the IBM PC. The type 1 keyboard with its 23-row by 4-column momentary contact pushbutton array is described in the SAMS Micro Maintenance series book IBM PC Troubleshooting & Repair Guide (22358). The type 2 keyboard design is described in the SAMS COMPUTERFACTS on the IBM PC 5150. Both designs function in a similar manner. They differ in the keyboard matrix configuration and in the way an on-board microcomputer is connected to the I/O cable. Because most systems today use the type 2 keyboard, this design will be described here.

Of the 17 components inside the keyboard chassis, 4 are ICs, 1 is a capacitive key matrix, and the rest are capacitors, resistors, and an inductor. Each of the 83 keys on the board is connected to a switch matrix. Each time you depress a key, you close a switch at a crossover point on an X-row by Y-column capacitive matrix. This matrix is scanned by an on-board processor that senses the open or closed contact condition of each crossover in the matrix. The closing of a switch causes the processor to generate a bit code stream that is passed through the keyboard cable onto the system board and into the 8255 PPI (U36) which sends the data on into the 8088 CPU (U3) for interpretation.

Each key is typematic in that depressing a key causes a scan code to be generated indicating a closed (make) condition. Releasing the key produces an open (break) scan code.

The heart of the keyboard circuitry is the 40-pin Intel 8048 8-bit microcomputer on-board processor (M1) shown in the block diagram of Fig. 2-56. The 8048 is a completely self-contained 8-bit parallel single-chip computer that contains a 1K by 8-bit program ROM and a 64 word by 8-bit RAM data memory. Twenty-seven I/O lines including two bidirectional ports, an 8-bit data bus, an 8-bit timer/counter, an on-board oscillator and clock circuits complete the architecture of this machine.

The pin assignments and pin utilization of microcomputer M1 are illustrated in Fig. 2-57. As shown, not all pins are used. The chip is configured to function as a key matrix monitor/controller that continuously scans the keys to recognize key action and generate a serial code that passes out SERIAL DATA output pin 17. It contains a self-test at power-on that checks its memory, and checks for stuck keys. M1 also maintains a bidirectional serial communications flow with the system board and executes the handshake protocol required for each scan-code transfer.

Table 2-15 lists the functions associated with the pins on the 8048 keyboard single chip computer (M1). Only the principal pins that are connected in the circuit are described.

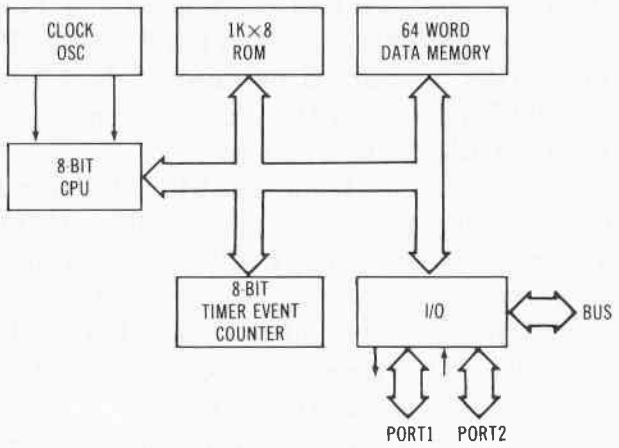


Fig. 2-56. 8048 8-bit microcomputer block diagram.

Single chip computer M1 contains an internal high gain parallel resonant circuit that uses pins 2 and 3 to connect a crystal or inductor-capacitor oscillator to generate the frequency reference for chip operation. Pin 1 is the input to the circuit amplifier stage. Pin 2 is the output of the stage. On the type 2 keyboard, a capacitor-

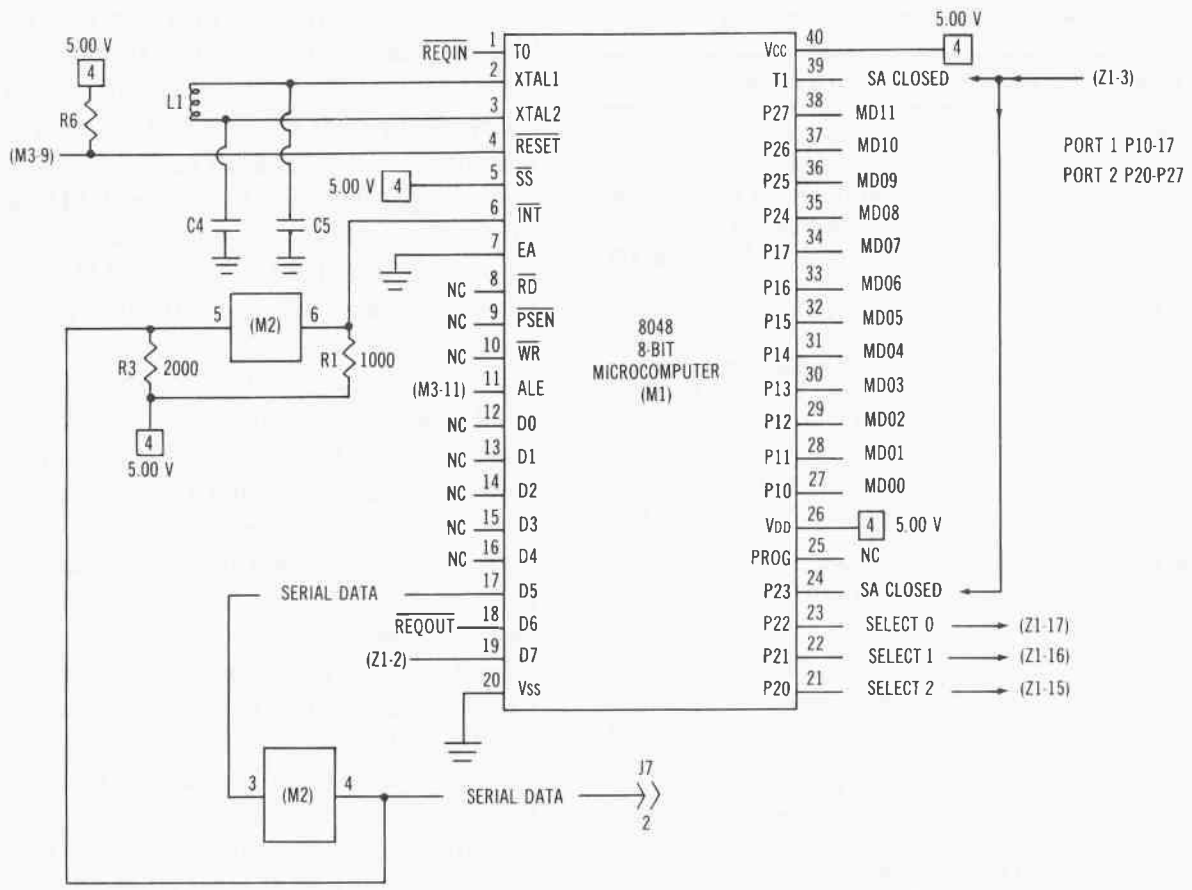


Fig. 2-57. 8048 8-bit microcomputer pin assignments.

inductor-capacitor circuit provides the feedback and phase shift required for oscillation because an accurate frequency reference and maximum (8 MHz) processor speed are not required to handle typing actions. The oscillator runs at approximately 4.77 MHz.

The output of the oscillator is divided internally by three to create a clock signal defining the state of the machine. This clock signal is made available on output pin 1 under program control. As illustrated in Fig. 2-58, the pin 1 output request in signal (REQIN*) is passed through keyboard buffer M2 onto the system board as a keyboard clock pulse used to clock the serial scan data coming out pin 17 of 8048 M1 off the keyboard circuit board onto the system board. The REQIN* clock pulse output is disabled by an address latch enable (ALE) pulse coming out pin 11 of M1. ALE is used to clock flip-flop M3, periodically resetting M1 so it can rescan the keyboard matrix.

The 12-row by 8-column keyboard capacitive matrix in Fig. 2-58 is scanned by M1 every 3 to 5 milliseconds. Twelve pins of 8048 M1 I/O ports 1 and 2 are connected to the keyboard matrix. Port 2 output pins 21, 22, and 23 form three binary weighted inputs to keyboard decoder/sense amplifier Z1. The three inputs (SELECT 0, SELECT 1, and SELECT 2) enter Z1 pins 15, 16, and 17 where they are decoded to activate a particular sense output (SENSE A through SENSE H) that connects to the keyboard capacitive matrix. The condition of the decoded sense line is returned to 8048 M1 via Z1 output pin 3 as signal SA CLOSED.

Therefore, when a key is depressed, this condition can be recognized by reading the input lines MD00 through MD11 and scanning the sense lines (SENSE A through SENSE H) to determine which intersection point is active. When a key closure is detected, the event timer program in M1 waits a few milliseconds to let the

Table 2-15. Pin Function Summary

Label	Pin(s)	Function
REQIN*	1	Request in: Active low input from buffer M2. Functions as external interrupt input. Low level on pin 1 causes subroutine jump at location 3 in program memory.
XTAL1	2	Crystal 1: One side of crystal input for internal oscillator. L1, C4, and C5 replace crystal in type 2 keyboard. Inductor L1 functions as the timing control element.
XTAL2	3	Crystal 2: Complementary side of external timing control element.
RESET*	4	Reset: Processor initialization input. The Q output from D-latch M3.
SS*	5	Single step: Tied high to keep the 8048 in the "run" mode.
INT*	6	Interrupt: Active low input from buffer M2. Initiates internal interrupt sequence.
EA	7	External access: Tied low to cause 8048 to make all memory fetches from internal memory.
ALE	11	Address latch enable: Occurs once during each cycle. Used as clock signal for D-latch (M3).
SERIAL DATA	17	Serial data: Part of bidirectional data bus. Used as output to buffer M2 to pass keyboard scan codes serially to system board.
REQOUT*	18	Request out: Active low output to buffer (M2).
SELECT 0-2	21-23	Select 0-2: Quasi-bidirectional port outputs that combine to generate a 3-bit input to 3-of-8 decoder/sense amplifier (Z1).
SA CLOSED	24	Sense amplifier closed: Active high input from sense amplifier (Z1) to indicate the closing of one of eight sense lines in the keyboard capacitive matrix. Also connected to input pin 39 (T1) to start internal event counter under program control.
MD00-MD11	27-38	Matrix data bus: 12-bit input bus connected to the keyboard capacitive matrix.
SA CLOSED	39	Sense amplifier closed: An input pin testable using 8048 instructions. Is used as the internal event counter input causing a delay before reading the matrix data bus input. This eliminates keybounce problems.

key bounce settle out and the key condition become stable before reading the MD00 through MD11 inputs. Then 8048 M1 stores the fact that a key closure has occurred in its internal RAM memory as an 8-bit scan code. If a key is held down longer than one-half second, M1 causes the same scan code to be generated ten times each second. Internal RAM enables M1 to buffer up to 16 key scan codes permitting type-ahead for the operator.

The release of a key closure is also detected by M1 causing the eighth bit (bit 7) of the scan code to be set high. This effectively produces a break code that is decimal 128 higher.

M1 also searches the array for phantom switch closures (several interconnections made and falsely encoded). If two keys are depressed causing closed switches in the same column, and one of the two rows containing a closed switch has another switch closed, a phantom switch condition has occurred. The 8048 single chip computer M1 usually ignores this condition. Only legitimate double and triple key closure operations are accepted. The scan is done in 3 to 5 milliseconds and at least 20 to 30 milliseconds pass between key entries, the matrix can be scanned at least once each keystroke and incorrect data encoding eliminated.

Each key action caused M1 to generate a unique scan code as shown in Table 2-16. A specific scan code is generated when a key is depressed and a scan code 128 higher is generated when the key is released. Depressing the Enter key causes M1 to generate the hex code 1DH (00011101 in binary). When your finger is removed from this key, 8048 M1 generates the code 9DH (10011101 in binary). Only the high bit has changed. This is the same as adding 128 to the original scan code. Once this code has been generated and passed out pin 17, the scan code signal drops to 0 (00H). Uppercase characters and special functions can be generated by depressing the Alt/Ctrl/Shift keys and one or more character keys.

When a key (or set of keys) has been depressed, M1 generates a logic high 0.2 millisecond signal that is passed out M1 pin 17 into pin 3 of the keyboard buffer (M2). The pin

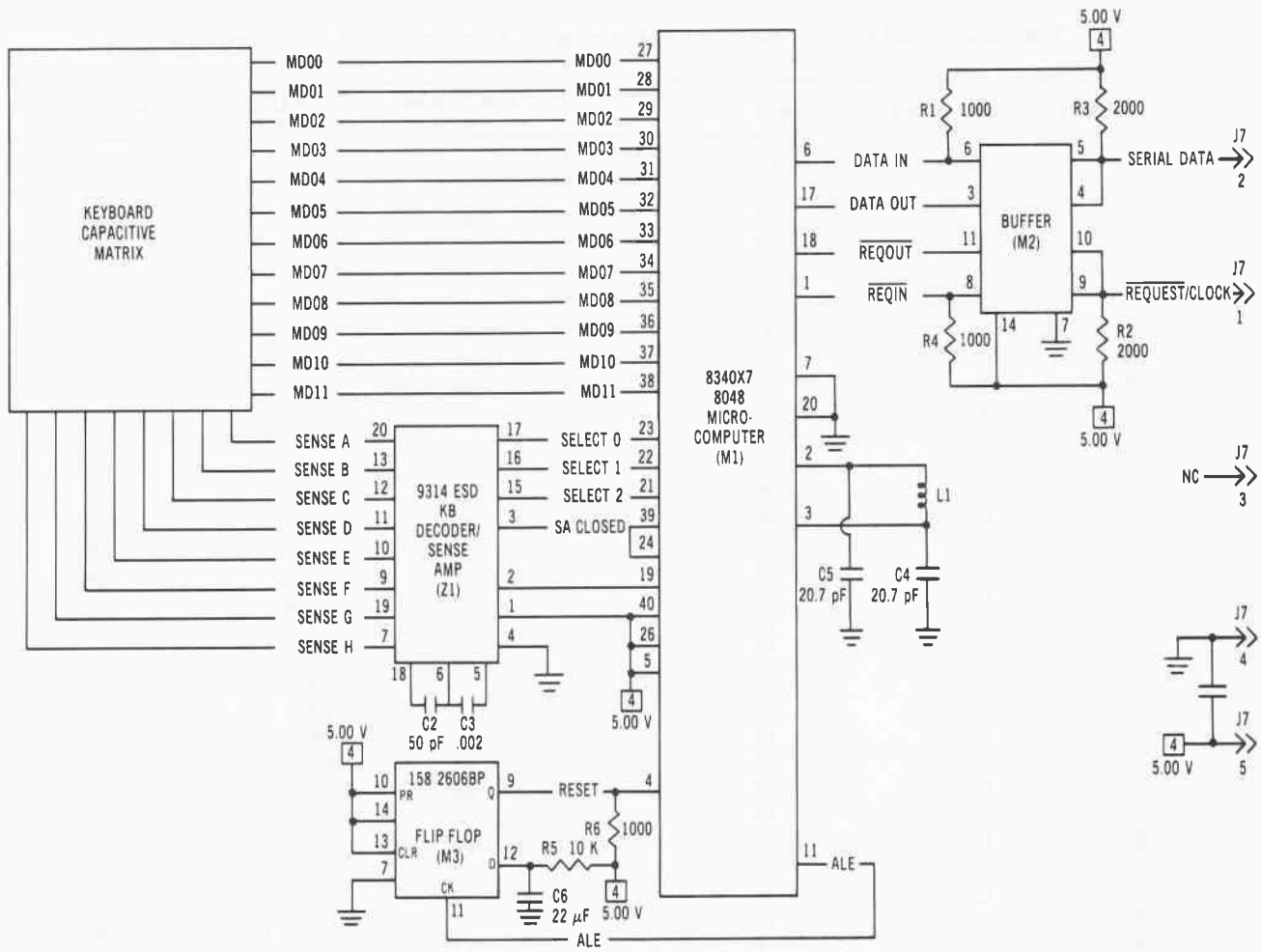


Fig. 2-58. Type 2 keyboard circuitry.

4 output of M2 notifies the system board circuitry that SERIAL DATA is coming. Then M1 clocks an 8-bit serial scan code through M2 out the keyboard and onto the system board (Fig. 2-59). This code is sent least significant bit first. Each bit is 0.1 millisecond wide.

The scan code data (SERIAL DATA) passes through the keyboard cable up to input pin 17 of 74LS322 serial/parallel buffer U24. In the meantime, 8048 M1 in the keyboard generates a request signal (REQUEST*) out pin 18 (REQOUT*) through buffer M2, out the cable and into pin 4 of 74LS175 quad-D flip-flop (U26). Flip-flop U26 is clocked by the 2.386363 MHz PCLK signal from the 8484 clock generator on the system board. The pin 6 output of U26 is a clock input on pin 11 of U24. This signal clocks the scan code data into pin 17 of U24. Pin 2 of U24 is tied high causing the chip to function in a serial-in/parallel-out configuration. Thus the scan code data is serially entered into U24 and then prepared for transmittal out as a parallel signal on the 8255 PPI (U36) port A bus (PA0 through PA7).

When the last of the 8-bit scan code has been shifted serially into U24, a signal is generated at pin 12. This signal is passed to the D input (pin 2) OF 74LS74 dual-D flip-flop (U82). On the next occurrence of a clocked output from 74LS175 quad-D flip-flop (U26) into pin 3 of U82, passing the state of U82 out its Q (pin 5), and Q* (pin 6) outputs. The Q* output becomes an enable signal to 74LS125 quad 3-state buffer (U80) pulling the keyboard SERIAL DATA in line low. The Q output becomes an interrupt request 1 (IRQ1) signal that enables the parallel output register of 74LS322 buffer (U24) so the 8255 PPI (U36) port A can receive

Table 2-16. Scan Codes Generated by Pressing the Keys on the IBM PC Keyboard

Key Number	Key Label	Scan Code	Key Number *"cont."*	Key Label *"cont."*	Scan Code *"cont."*
1	Escape	01			
2	1	02	43	\	2B
3	2	03	44	z	2C
4	3	04	45	x	2D
5	4	05	46	c	2E
6	5	06	47	v	2F
7	6	07	48	b	30
8	7	08	49	n	31
9	8	09	50	m	32
10	9	0A	51	<	33
11	0	0B	52	>	34
12	-	0C	53	/	35
13	=	0D	54	Shift	36
14	Backspace	0E	55	Pt Sc	37
15	Tab	0F	56	Alt	38
16	q	10	57	Space	39
17	w	11	58	Caps Lock	3A
18	e	12	59	F1	3B
19	r	13	60	F2	3C
20	t	14	61	F3	3D
21	y	15	62	F4	3E
22	u	16	63	F5	3F
23	i	17	64	F6	40
24	o	18	65	F7	41
25	p	19	66	F8	42
26	[	1A	67	F9	43
27	]	1B	68	F10	44
28	Enter	1C	69	Num Lock	45
29	Ctrl	1D	70	Scroll Lock	46
30	a	1E	71	7	47
31	s	1F	72	8	48
32	d	20	73	9	49
33	f	21	74	-	4A
34	g	22	75	4	4B
35	h	23	76	5	4C
36	j	24	77	6	4D
37	k	25	78	+	4E
38	l	26	79	1	4F
39	;	27	80	2	50
40	,	28	81	3	51
41	'	29	82	0	52
42	Shift	2A	83	Del	53

PA0 through PA7. IRQ1 is also passed to pin 19 of the 8259 programmable interrupt controller (U2).

Upon receipt of the IRQ1 interrupt request, U2 generates an interrupt signal (INT) that alerts the 8088 CPU (U3) that an external device wishes to communicate with it. (This process was covered in the section on interrupts.) CPU U3 halts what it is doing and responds by passing a code out its S0 through S2 lines into the 8288 bus controller (U6). U6 reacts by generating an interrupt acknowledge (INTA) that is passed back to U2. As described in the section on interrupts, a second INTA is generated causing the 8259 (U2) to place an interrupt code (INT 9) on the system data bus.

The 8088 CPU (U3) reads the data bus and calls the INT 9 subroutine in the ROM-based operating system. INT 9 causes the scan code to be read into port A (PA0 through PA7) of U36

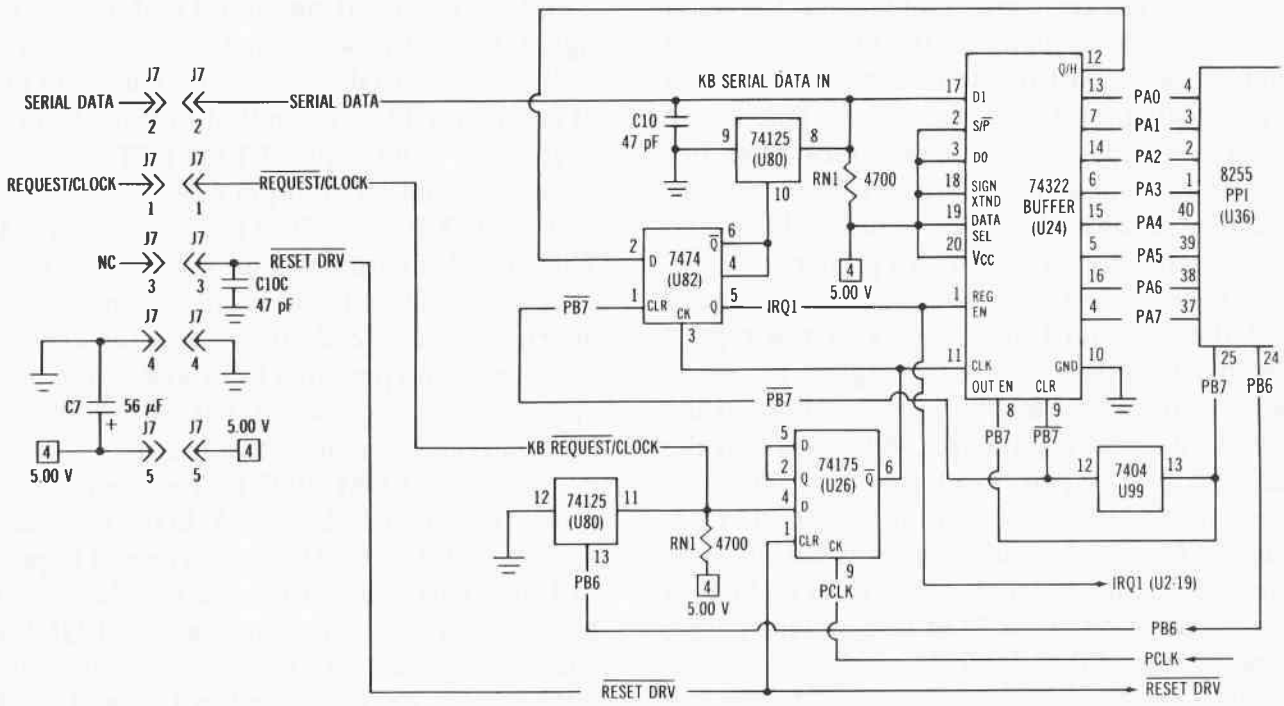


Fig. 2-59. System board keyboard interface.

and on into the 8088 CPU (U3). The scan code is converted by U3 into an ASCII code for the character selected. An ASCII conversion chart can be found in Appendix G. The 8-bit scan code and the 8-bit ASCII character code are stored in 32 consecutive locations of a 16-character circular RAM buffer. INT 9 also clears the interrupt request by causing the PPI (U36) to generate an active high PB7 signal out pin 25 of U36 where it gets inverted by 74LS04 hex inverter (U99) to produce PB7*. This active low signal clears the 74LS74 dual-D flip-flop (U82) removing the IRQ1 active high signal so another interrupt can occur.

The ASCII character and the scan code are combined as two bytes per character with the low byte representing the encoded character and the high byte the keyboard scan code. For example, the ASCII code for lowercase "a" is 097 decimal (61H). The scan code generated when "a" is depressed is 1EH. Therefore, the keyboard buffer in RAM contains 611EH (0110 0001 0001 1110 in binary). Special keys such as a function key or numeric keypad key usually have the scan code in the high byte and all zeroes in the low byte. The buffer content for depressing function key F1 (key position 59) would, therefore, be 3BH yielding a buffer code 3B00H.

Keystrokes are captured and translated by the basic input/output system code (BIOS) stored in the ROM chips on the system board. The system's ROM BIOS produces an INT 16 interrupt causing the ASCII and scan codes to be read out of the RAM buffer. INT 16 causes the operating system software to execute a keyboard I/O routine. This routine reads through the keyboard buffer until it finds a character code. The value of this 16-bit word is read into the 8088 (U1) accumulator register (AX). The scan code stays in the upper byte (AH). If this byte is all zeroes, a noncharacter key has been pressed. The program then reads the status of the data to determine if any special keys such as Ctrl, Alt, or Shift were depressed. The ASCII character code is passed to the calling program which uses the character in a software code character string or data input while the BIOS continues its program execution, passing the character to an active output device (screen or printer) to complete the operator-machine-operator interaction or uses it internally as a special password.

Some keys, or key combinations, are directly affected by the ROM BIOS. The Alt key can be used to directly enter ASCII character

codes from the keyboard. Holding this key down and typing three digits on the numeric keypad enables the ROM BIOS to recognize the input as representing an ASCII character. This is one way to generate graphic characters. Another built-in function key is */Prtsc. Depressing shift and */Prtsc causes the BIOS to print all the characters on the display screen on a printer.

Communication between the system board and the keyboard also occurs in the type 2 keyboard configuration. Two signals are generated by the ROM BIOS to reset or to clear the keyboard circuitry. Figure 2-60 shows that the 8255 PPI (U36) is used to send software-generated commands from the 8088 CPU (U3) into the keyboard circuitry. Port B of U36 is the gateway for these signals. Port B bit 6 (PB6) is connected between pin 24 of U36 and the enable input pin (pin 13) of 74LS125 quad 3-state buffer (U80). Pin 25 of U36 connects PB7 to pin 8 of 74LS322 serial-to-parallel shift register (U24), and to pin 13 of 74LS04 hex inverter (U99).

One section of code in the ROM BIOS causes the binary code 0 0 0 0 1 1 0 0 (hex 0CH) to be placed in the lower 8 bits of the 8088 CPU accumulator register (AL). The contents of AL are then passed to port B of U36. Then, PB6 and PB7 are both zero. The low PB6 output from pin 24 of U36 enables U80 generating an active low REQUEST* signal out pin 11 of U80. This signal holds the keyboard clock low for 20 milliseconds. REQUEST* is also passed to pin 1 of connector J7 into pin 9 of buffer M2 in the keyboard. The input REQUEST* signal is relabelled REQIN* out pin 8 of M2 and passed into pin 1 (T0) of the 8048 microcomputer (M1). This signal functions as an external interrupt causing the 8048 to do a subroutine jump to internal program code that resets the keyboard electronics and prepares it for key action by the operator. Scan code AAH is returned to the 8088 CPU (U3) on the system board.

Another ROM BIOS subroutine places the code 1 1 0 0 1 1 0 0 (hex CCH) in the internal AL register of U3. The AL is again passed to port B of U36. This brings PB6 and PB7 both active high. Buffer U80 is not enabled so REQUEST* CLOCK is held high. However, the high signal out PB7 (pin 25) is inverted by U99 and used to clear 74LS74 dual D latch (U82). This brings the Q* output (pin 6) of U82 low enabling a section of 74LS125 quad 3-state buffer (U80) passing a low signal out pin 8 through pin 2 of connector J7 and into pin 5 of buffer M2 in the keyboard. This low becomes the CLEAR DATA signal that enters pin 6 (INT*) of 8048 microcomputer (M1). Active low on pin 6 causes M1 to generate an internal interrupt clearing the keyboard.

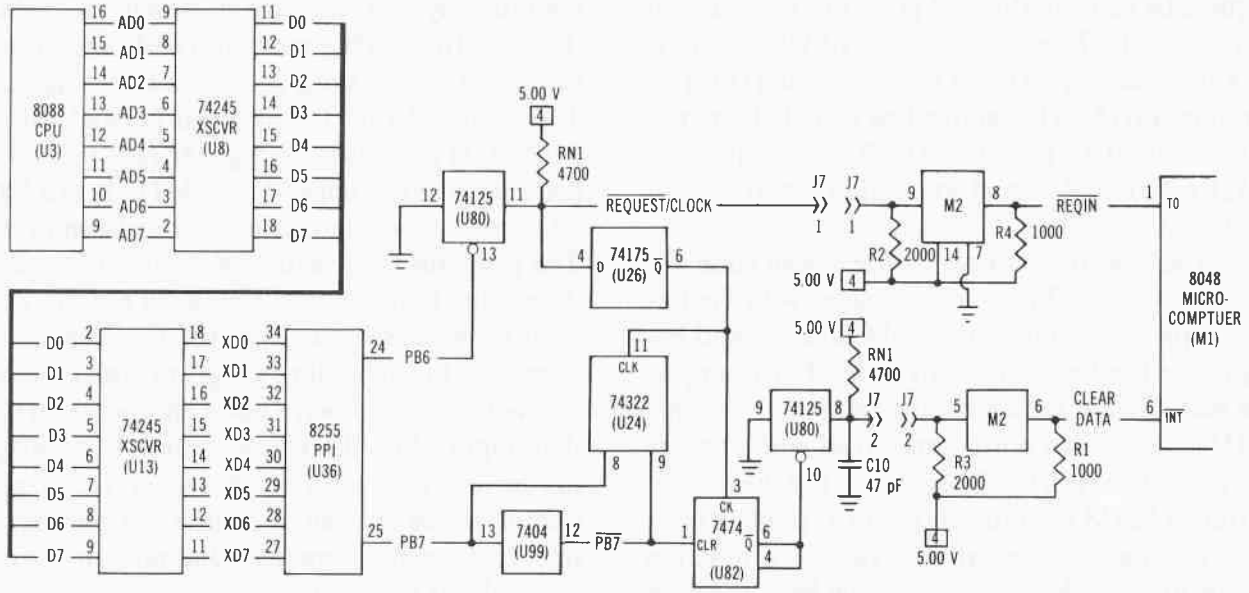


Fig. 2-60. Keyboard RESET and CLEAR circuitry.

8087 NUMERIC PROCESSOR EXTENSION (NPX)

The 8088 CPU in the IBM PC is designed for general data processing applications requiring fast, efficient data movement and control instructions. The arithmetic performed in this microprocessor is simple. However, the PC is being used in many engineering, scientific, and business batch processing applications. These environments require more powerful arithmetic instructions and data types and deal with fractional values, sine, cosine, logarithms, and square roots. The results of complex operations require extreme accuracies and rapid calculations. The Intel 8087 numeric processor extension (NPX) was designed to support these operations while residing in the same circuitry as the 8088 CPU. Both the hardware and the software interfaces for the 8087 are compatible with the 8088.

The 8087 NPX is ideal for number crunching applications. It can handle numeric data over a wide range of values including integer and noninteger values. The 8087 can produce very precise results with a large number of significant digits. It can also handle calculations involving real and imaginary numbers. Integers up to 18 digits long and floating point numbers from 16 to 80 bits wide can be used in this machine. The 8087 can respond to built-in math instructions for add, subtract, multiply, divide, absolute value, arctangent, tangent, square root, log base two and log base e.

A prewired 40-pin socket is mounted on the system board to hold an optional 8087 NPX. With the 8088 CPU configured in maximum mode, the 8087 NPX shares the multiplexed address/data bus, S0 through S2 status signals, queue status signals, ready status signals, clock, and reset. The pin assignments for the 8087 NPX are shown in Fig. 2-61. Two output signals, test (TEST*) and numeric processor numeric processor interrupt (NP NPI) on pins 23 and 32 respectively are used to inform the CPU of NPX internal status. The corresponding Intel labels are BUSY and INT. To ensure that the 8088 CPU always sees a "not busy" status if no 8087 NPX is installed, the input to CPU U3 is connected to the 5 volt supply through a 4700 ohm resistor in resistor network RN1.

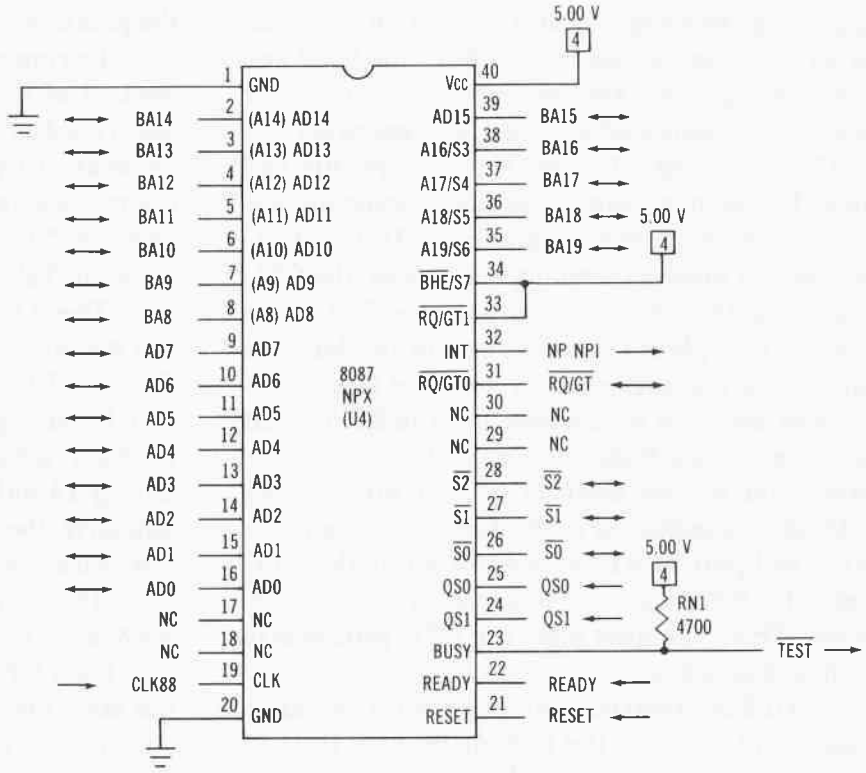


Fig. 2-61. 8087 Numeric processor extension (NPX) pin assignments.

The TEST* (BUSY) signal is designed as an active high signal to tell the CPU that the NPX is executing a numeric instruction. The TEST* input to pin 23 of the CPU is designed as a test input that is sampled by a wait instruction. If the input is low, CPU execution continues, otherwise the 8088 goes into an idle state. The Intel hardware reference manual on the 8087/8088 suggests installing a 10K pull-down resistor on the BUSY/TEST signal line between the NPX and the CPU sockets to ensure that the CPU always sees a "not busy" status if no 8087 is installed on the system board. With the 4.7K pull-up resistor on the IBM PC system board, and no 8087 NPX installed, a CPU wait instruction will interpret the active high input on pin 23 as an indication that a numeric instruction is in process within a (phantom) coprocessor. This will cause the CPU to enter an idle state forever. Therefore, a wait instruction should not be used unless an 8087 NPX is actually installed in the system.

The 8087 NPX is designed to sit idle and monitor the 8088 CPU instructions. Both the CPU and the NPX decode the same instructions. Only the CPU can respond to CPU instructions. A numeric instruction that requires 8087 action appears as an escape (ESC) instruction. As soon as an ESC instruction occurs, both the 8087 and the 8088 decode and execute it. A numeric operation begins when the CPU executes the ESC instruction. The NPX monitors this and looks for the next byte of the ESC instruction and acts on this coded command. If the ESC instruction involves memory references, the CPU calculates the effective address specified in the instruction, places the address on the bus and initiates a memory read cycle. The 8088 CPU ignores the data it receives, but the 8087 reads the data coming from memory and completes the execution of the instruction. If no memory reference is indicated in the ESC instruction, the NPX will pull TEST* active low idling the CPU until the NPX numeric instruction is complete. When TEST* returns high, the CPU proceeds to its next instruction.

All ESC instructions begin with the most significant 5 bits of the high order byte (bits 11 through 15) a logic 11011. The most significant 2 bits of the low order byte (bits 6 and 7) define the addressing mode and the number of bytes remaining in the instruction. The nonmemory form of ESC causes specific coprocessor activity based on the nine remaining bits. The 8088 CPU ESC instructions provide 64 memory reference opcodes and 512 nonmemory reference opcodes. The 8087 NPX can respond to 57 of the memory reference opcodes and 406 of the nonmemory reference opcodes.

Coprocessor instructions are labelled beginning with F for "Floating point." Therefore, FMUL represents a floating point multiplication instruction.

To ensure that the 8087 is finished with its operation before the CPU should fetch and execute the next instruction in the program, a wait instruction precedes the ESC. If the 8088 must wait for the 8087 to complete storing data in memory before CPU operations can continue, the programmer places another wait instruction in the code just after the ESC instruction. In this way, the wait and ESC instructions work together to enable CPU and NPX operations on the same system board hardware.

Table 2-17 describes the signals connected to the pins of 8087 NPX (U4).

During operations when the CPU (U3) has control of the bus, the eight status line signals described in the section on the 8088 CPU are generated by U3. During NPX bus control operations, only five combinations of the status lines (S0* through S2*) signals are encoded, as shown in Table 2-18.

These status signals are driven active during bus machine cycle T4. They remain valid during T1 and T2, and they shift to a passive state (1,1,1) during clock tick T3 or during TW when READY is high. Any change in S2*, S1*, or S0* during T4 indicates the beginning of a bus cycle. Similarly, the return to the passive state in T3 or TW marks the end of a bus cycle.

The primary interfaces between the 8087 NPX and the 8088 CPU are shown in Fig. 2-62. The 8088 CPU (U3) and 8087 NPX (U4) fetch the same instructions and bytes of the instruction stream. This allows the 8087 to monitor and decode instructions synchronously with the 8088

Table 2-17. Pin Signal Descriptions

Label	Pin(s)	Type	Name and Function
BA15-BA7	2-8, 39	I/O	Buffer address: The time multiplexed high order memory address byte.
AD7-AD0	9-16	I/O	Address bus: The time multiplexed low order memory address byte.
BA19-BA16	35-38	I/O	Buffered address: Address lines during T1 of memory operations. Input lines to monitor the 8088 when it has control of the bus.
CLK88	19	I	Clock 88: A 4.772727 MHz asymmetric 33% duty cycle clock to provide optimized internal timing for the processor and bus controller.
RESET	21	I	Reset: An internally synchronized signal that causes NPX to end present activity. Must be high for at least four clock cycles.
READY	22	I	Ready: Active high acknowledgment from addressed memory device that it will complete data transfer. READY is synchronized by 8284 clock generator (U11).
TEST*	23	O	Test: When active high, indicates to 8088 (U3) that 8087 NPX is executing numeric instruction. When TEST* is low, U4 is idle.
QS0, QS1	24, 25	I	Queue status 0, 1: Status inputs from 8088 CPU to allow 8087 NPX to track CPU instruction queue.
S2*-S0*	26-28	I/O	Monitor inputs when 8088 has control of bus. Status outputs when 8087 is in control.
RQ*/GT*	31	I/O	Request/grant: Used by 8087 to request the local bus. 8088 CPU generates grant and release pulses.
NP NPI	32	I	Numeric processor interrupt: An active high output signal used to indicate that an unmasked exception has occurred during numeric instruction execution when 8087 interrupts are enabled. This signal is routed to the NMI circuitry.

CPU without introducing any CPU overhead. Thus, U4 knows at all times what instruction was just fetched into U3. The NPX and CPU instructions can be mixed in the program instruction stream. And the 8087 NPX can process instructions in parallel with and independent of the 8088 CPU.

Table 2-18. 8087 NPX Bus Cycle Status Line Encoding

S2*	S1*	S0*	Function
0	X	X	Unused
1	0	0	Unused
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive

When the 8087 is executing a numeric instruction, the TEST* line (pin 23) is used to resynchronize the system. The 8088 WAIT instruction tests the TEST* input to determine when the NPX can execute later instructions.

The NPX (U4) maintains an instruction queue that is identical to the CPU. Its queue length is automatically set to match that of the CPU by the continuously active high input on pin 34 (BHE*/S7 in Fig. 2-61). Therefore, immediately after reset, the queue length in U3 and U4 are identical.

To help coprocessor U4 in monitoring the CPU operations, nine status lines are connected between the 8087 and 8088 machines (QS0, QS1, and S0* through S6* as shown in Fig. 2-61). The QS0 and QS1 inputs on U4 pins 25 and 24 respectively are used to monitor the CPU and enable the NPX to obtain and decode instructions synchronously. Table 2-19 describes the decoded meaning for the combination QS0 and QS1 inputs from CPU (U3).

Table 2-19. QS0-QS1 Code Definitions

QS1	QS0	Description of CPU Status
0	0	No operation
0	1	First byte of op code from queue
1	0	Empty the queue
1	1	Next byte from queue

Notice that the status signals S0* through S2* are also passed to the 8288 bus controller (U6) in the lower right of Fig. 2-62 to generate the memory and I/O access commands, and to the 74530 8-Input NAND (U5) where they combine with the LOCK* signal from U3 and generate an output for the HOLDA circuitry. HOLDA is used by the 8237 DMA controller (pin 7 on U35) and by 74LS175 quad D-latch (U98) which generates the AEN* and AEN BRD signals described earlier in the chapter.

The 8087 NPX (U4) can interrupt CPU (U3) whenever it detects an error or exception. If an incorrect instruction is sent to U4, such as a command to load a full register, or to divide a number by zero, U4 can signal the CPU (U3) via an interrupt. The 8087 interrupt circuitry is shown in Fig. 2-63.

Without an 8087 NPX installed in socket U4, switch SW1 position 2 is opened (OFF position) pulling the pin 12 input to 74LS00 quad 2-input NAND gate (U81) high. When an 8087 NPX is installed in the system, SW1-2 is closed (ON position) grounding the pin 12 input to U81. The other input to U81 is the NP NPI line from the empty 8087 NPX (U4) socket. NAND gate U81 output (pin 11) connects to one of the three inputs to 74LS10 triple 3-input NAND gate (U84). The pin 10 input to U84 comes from the Q* (pin 6) output from 74LS74 dual D-latch (U96). On the positive edge transition of the pin 3 clock input XMEMR*, U96 latches the D input (pin 2) into the flip-flop. Pin 2 connects to the output (pin 10) of 74LS02 quad 2-input NOR gate (U27). The pin 8 and 9 inputs to U27 are (RAM ADDR SEL)* and the EVEN parity output from pin 6 of the 74LS280 parity generator/checker (U94).

The third input to the 74LS10 triple 3-input NAND gate (U84) comes from pin 8 of 74LS00 quad 2-input NAND gate (U52). U52 is enabled

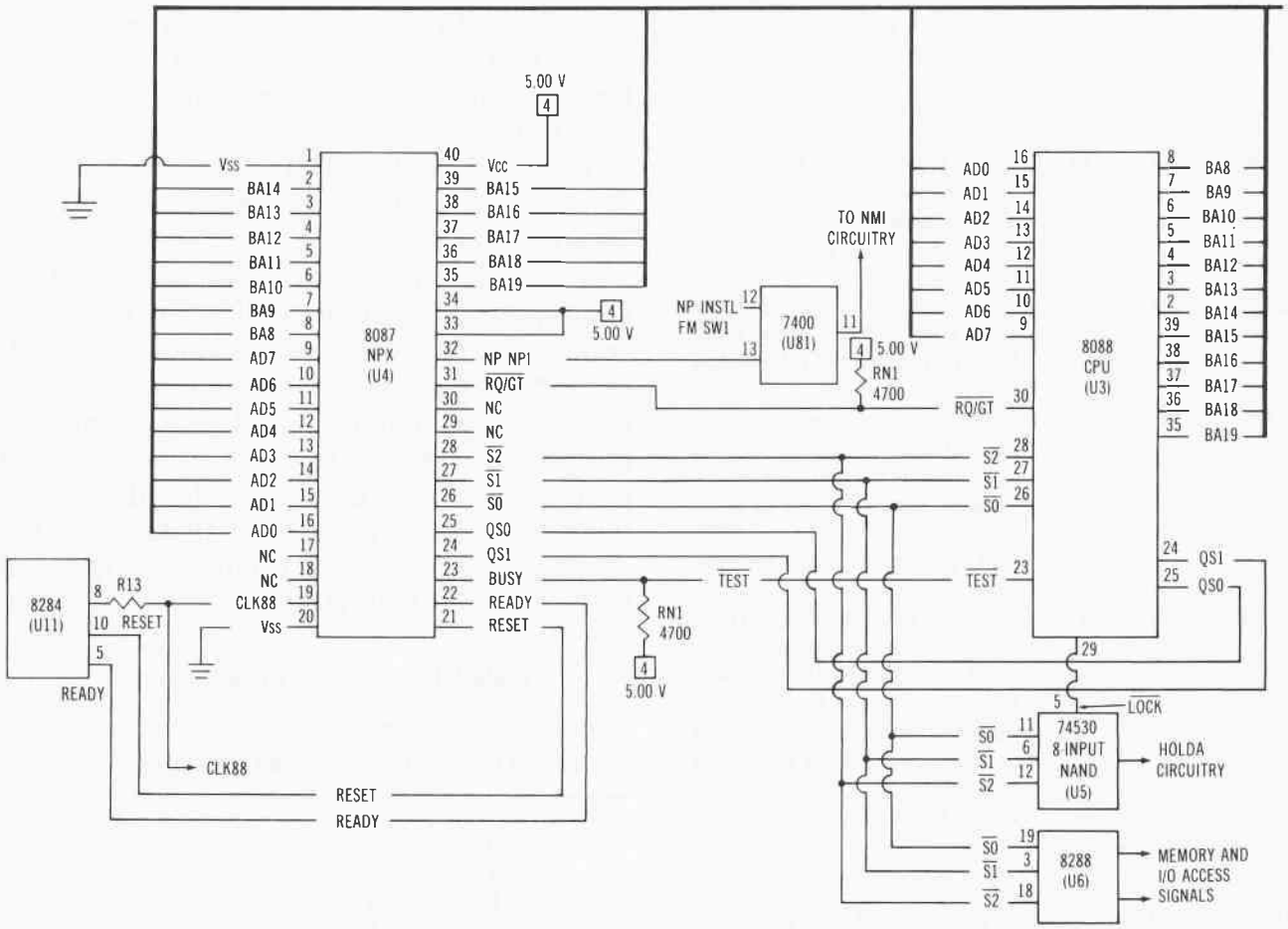


Fig. 2-62. 8087 NPX (U4) and 8088 CPU (U3) interface circuitry.

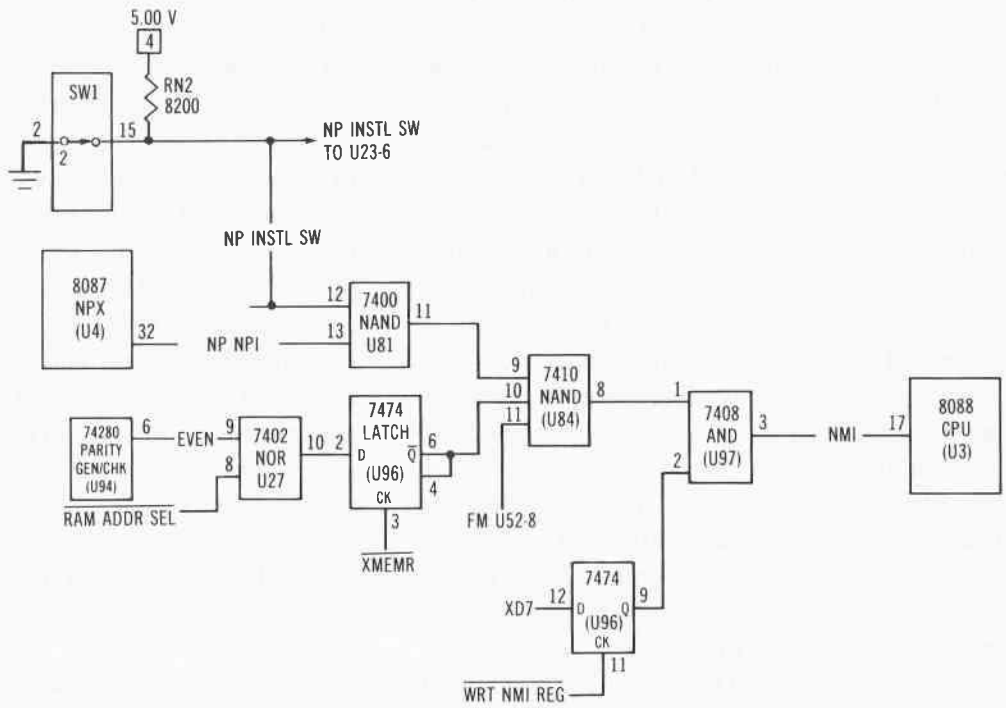


Fig. 2-63. 8087 NPX (U4) interrupt circuitry.

by I/O CH CLK and EN I/O CK. The pin 8 output from U84 becomes an input (pin 1) to 74LS08 quad 2-input AND gate (U97). The other input to U97 (pin 2) is the Q output from the other half of 74LS74 quad D-latch (U96). This flip-flop follows the XD7 input (pin 12) whenever (WRT NMI REG)* transitions from active low to high. The ANDed output from U97 (pin 3) is the non-maskable interrupt (NMI) described earlier in the chapter. NMI is passed directly into pin 17 of the 8088 CPU (U3).

With an 8087 NPX installed in socket U4, only three conditions will prevent an interrupt from occurring once interrupt conditions exist inside U4.

1. Certain bits in the 8087 control word (exception and interrupt enable) are set high.
2. System board switch SW1-2 is closed (ON position).
3. The 74LS74 NMI mask register (U96) is set to ONE.

U96 is cleared at power-up to disable NMI. Once quiescent conditions are met in the system circuitry, U96 is free to be set by clocking (WRT NMI REG)* on pin 11 to place the logic condition of XD7 in the U96 flip-flop.

Intermittent 8087 System Performance Problems

The 8087 NPX draws 0.457 milliamp of current and dissipates up to 3 watts. This can create two related problems: increased heating inside the chassis, and excessive circuit power consumption. If the system has several power-hungry expansion boards, the standard 67.5 watt power supply can be stressed worsening the heat problem. These thermal management issues can cause intermittent CPU lockup, power-on self test failures, suspicious RAM memory malfunctions, unreliable disk drive write operations, and even power supply failure. Additional cooling and/or increased power supply capability may be required to prevent 8087-caused system failures.

VIDEO

Three types of video signals are produced on interface boards plugged into expansion slots in

the PC system board. One signal is NTSC composite video comprised of horizontal and vertical syncronization signals combined with the video information. For a color display, the composite signal has color information phase and amplitude modulated as a 3.579 MHz subcarrier superimposed on the standard composite video. A second type of video is an RF amplitude modulated signal that can drive a standard television receiver. The third type signal is comprised of discrete video and synchronization signals that can directly drive a special IBM display monitor. All three of these signals will be described in this section.

A video screen image is generated as thousands of tiny picture elements (pixels) arranged in neat rows and columns. Each row is called a "raster," and each column position in a raster is called a "pixel" or "dot."

Each display unit (monitor or television) has a cathode ray tube (CRT) inside. The face of the CRT is what you observe when you look at a display screen. A beam of electrons from the inside rear of the cathode ray tube (CRT) inside the display unit scans across the screen under control of a horizontal sawtooth wave signal. As the beam moves across the screen, its intensity may increase momentarily as it passes over specific pixel positions on the screen. Upon reaching one side of the screen, the electron beam is essentially stopped and the focus point is retraced back to its original column. However, a second control signal is applied to vertical deflection plates inside the CRT to move the beam down one row. Therefore, the beam is moved horizontally by a horizontal sweep signal, blanked out during retrace, and shifted vertically down a raster by the vertical sweep signal. When the last raster row has been traced, the beam is blanked and the CRT focus returned to the upper left corner of the display area.

The inside of the CRT screen is coated with a material that glows when hit by high intensity electron beams. The increased intensity dots form patterns on the screen that we recognize as characters or graphic shapes. The pattern of intensity is determined by a sequence of binary values generated within the adapter board connected to the PC system board. These binary values are synchronized to the horizontal and vertical sweep frequencies of the display and are transmitted to the display unit via one of several video connectors on the board. Each logic high in the dot pattern causes a particular pixel position on the inside of the CRT screen to receive a high intensity beam of electrons from the CRT gun. Raster displays of picture or text are produced when this beam of electrons is repeatedly scanned across the inside surface of the CRT screen to form a pattern of closely spaced horizontal lines (the raster) which covers the entire screen. The inside surface of the screen is covered with a phosphor that glows when the electron beam hits it, so pictures are formed as the beam turns on and off while scanning along each raster line on the screen surface.

The number of raster lines and dot pixels on each raster defines the resolution of the display. The more raster rows and the more pixels per raster, the higher the resolution.

All the video signals required by the display monitor are produced on external interface boards. Address, data, and control signals are passed from the system board onto the expansion bus backplane and into one of many types of video display boards designed for the IBM PC. Only one video adapter board is required, although one can use a color and a black-and-white display board in the same system to provide flexibility. The two display boards produced by IBM are the monochrome monitor/printer adapter and the color/graphics monitor adapter. Because non-IBM display boards are also available, this text only provides an overview of the IBM monochrome and color/adapter cards. The ideas described for these two boards apply to the adapters provided by other companies.

The IBM monochrome display/printer adapter can produce 350 raster lines of vertical resolution. Each raster line has 720 dots of horizontal resolution. The IBM color/graphics monitor adapter can produce 200 raster lines of vertical resolution and 640 dots of horizontal resolution. These pixel specifications define a

screen matrix area 350 x 720 or 200 x 640 dots in size. Each of these matrices can easily fit within the 525 lines of horizontal resolution in a standard television receiver.

Synchronization signals sent to the display unit control horizontal and vertical sweep oscillators that electrically make the CRT electron beam scan across the screen. Horizontal sync pulses lock the picture horizontally to the incoming signal. Vertical sync pulses perform a similar role in the vertical circuitry of the display unit and prevent the display picture from rolling in a vertical direction.

Coarse and fine adjustments control the relationship of the video dot pattern information to the horizontal and vertical sweep of the CRT gun. Coarse adjustment is made using the horizontal and vertical hold controls on the outside of the monitor. These controls affect the oscillation frequencies of the CRT electron beam circuits. Fine adjustment is accomplished using horizontal and vertical sync signals passed with the video dot pattern information to the display unit from the adapter board.

Video monitors have wider signal bandwidths (10 to 20 MHz) than a standard television (4.5 MHz) so they can display small details clearly. In a standard television, the horizontal sweep frequency is 15,750 Hz, so each scan line requires about 53.5 microseconds plus 10 microseconds for the horizontal retrace. The vertical sweep frequency is 60 Hz, which allows 262.5 horizontal scan lines during one vertical sweep period. It takes 16.7 milliseconds (one-sixtieth of a second) to scan the entire screen area once (this is called one field), and 60 fields are displayed per second. Only about 245 of the horizontal lines are visible on the screen, however, since 1.25 milliseconds is required for the vertical retrace that brings the beam back to the top of the screen after a field has been completed.

The IBM monochrome display monitor uses a higher than normal 18.432 kHz horizontal sweep frequency and a lower than normal 50 Hz vertical sweep frequency so 350 raster lines can be traced within one complete scan of the screen. The monochrome adapter produces 720 dots for each horizontal raster line to obtain the 80 characters per row. These dots must be transmitted from the adapter board to the display unit during the time required to scan a single raster line. This places severe performance requirements on the circuitry in the display and the adapter. The dot transmission rate requires a video bandwidth of more than 16 MHz for the video amplifiers in the display. Because of the high bandwidth and nonstandard sweep rates, only IBM monitors and a few non-IBM monitors can be used with the monochrome adapter card. The monochrome adapter card provides a single direct drive video interface to the display unit.

The color monitor used with the PC (color/graphics card installed) has a horizontal sweep rate of 15.74 kHz and a vertical drive frequency of 60 Hz. This enables up to 640 pixels and 200 rows of display dots with a video band-width of 14 MHz. The color/graphics adapter card provides direct drive RGB video for high resolution monitors and composite video for low resolution (monochrome or color) monitors and a television receiver via an RF modulator connection.

The characters (numbers and letters) or shapes that are drawn on a computer display are generated by turning on or off an electron beam in the display unit as the beam sweeps (scans) across each raster line producing a pattern of bright dots on the screen. The characters or graphic shapes are represented by closely spaced dot patterns that are drawn one raster line at a time.

The generation of color video works the same way as monochrome except that, with color, three electron beams are used and the phosphor screen is coated with a pattern of circular or rectangular dots arranged in a three-pixel configuration. One of the three dots glows red when hit by an electron beam, another dot glows green, and the third dot glows blue.

Characters are produced in a fixed matrix within the dot array of the complete screen. The monochrome display adapter produces letters within a 7 by 9 dot matrix contained within a 9 by 14 character box. The extra lines (rows) and columns of dots allow room for descenders and for intercharacter and interline spacings.

Monochrome and color video are produced in different ways, but each type of IBM display adapter uses a Motorola 6845 CRT controller (CRTC). The 6845 CRTC will be described in the section on the monochrome monitor/printer adapter. Its operation is similar on the color/graphics adapter card. Refer to COMPUTER-FACTS CSCS2-A for this discussion.

Monochrome Monitor/Printer Adapter

The monochrome monitor/printer adapter generates black-and-white video and printer signals on a common circuit card. This text will cover the video portion of the board only. The direct drive video output from a 9-pin D-connector at the rear of the card produces 720 dots of horizontal raster lines on a connected monochrome display monitor. The vertical sweep frequency generates 350 raster lines on the display. This enables a 25-row display with 80 characters per row.

The adapter generates video characters only. No graphics are produced on this board. Each character is produced as a 7 by 9 dot matrix within a 9 by 14 dot array. The extra dots around the character provide spacing between characters and lines and provide room for letters with descenders (such as the letter g).

As described earlier, the horizontal and vertical sweep frequencies generated on this adapter card are not standard. A higher-than-normal video bandwidth requirement and nonstandard sweep rates limit the display options to few monitors except the IBM product. In addition, the video output is not a composite signal with the horizontal and vertical sync signals combined with the video dot pattern. Instead, the signals present on the pins of the output connector provide direct drive video as shown in Fig. 2-64. The video is comprised of separate sync signals, a video dot pattern signal, and an intensity signal that can produce bright, boldface letters. The "custom" video output configuration reduces the video display unit selection.

6845 CRT Controller (CRTC)

The 6845 CRT Controller (U35 in folder CSCS2-A) is an LSI component designed to provide an interface for the 8088 CPU to a raster scan CRT display. The primary function of the 6845 is to generate timing signals necessary for raster scan displays under the control of the 8088. The design of U35 places keyboard functions such as cursor movements, editing, and read/write under control of the system board CPU (U3) while providing the video timing and refresh memory addressing signals. U35 is fully programmable via the 8088 data bus. Under software control it generates a variety of character widths, display modes, and graphic shapes.

The chip can be programmed to generate a fixed height cursor, blinking characters, and interlace or noninterlace scan. The noninterlace scan is used in the IBM PC display system. As shown in Fig. 2-65, U35 is a 40-pin compact IC. Eleven of the 14 refresh memory address lines are used in the IBM PC design. Four pins are used to generate an address into the character generator (U33). The 8-bit data bus is buffered off the expansion board backplane into U35 as buffered data bits BD0 through BD7. A reset pulse is applied on pin 2. On the lower right of Fig. 2-65, five control inputs are used to select, enable, and clock the chip. Besides the refresh memory address and character generator row address outputs, U35 generates a horizontal and vertical sync, cursor, and display enable signals.

Table 2-20 describes the functions of each data, address, or control pin of CRTC U35.

The monochrome display/printer adapter and the color/graphics monitor adapter boards are both designed around the 6845 CRT controller (CRTC). A block diagram of this chip is shown in Fig. 2-66. The 6845 CRTC control chip is used to generate refresh addresses over MA0 through MA10, character generator raster addresses over RA0 through RA3, a display enable, and video monitor timing signals HSYNC and VSYNC. The 6845 CRTC also has an internal register that generates an output CURSOR signal whenever its contents match that of the current refresh address. A light pen

Table 2-20. 6845 Pin Description

Signal	Pin	I/O	Function
VSS	1	I	Ground
RESET*	2	I	RESET* Active low input that clears all internal counters and stops display operation. Forces all outputs except the data bus to a low condition. Does not affect internal control registers.
LPSTB	3	I	Light pen strobe: High impedance input that is synchronized by the clock input on pin 21 to latch current refresh memory address (MA0-MA13) in 14-bit light pen register on the low to high transition of a strobe pulse detected by the light pen and control circuit.
MA0-MA10	4-14	O	Memory address (refresh): Active high outputs that enable 2K words of refresh memory (one 2000 character screen). (NOTE: MA11-MA13—pins 15 through 17 are not used.)
DISPEN	18	O	Display enable: Active high signal indicating the CRTC is providing addressing in the active display area. This signal defines the display period in horizontal and vertical raster scanning. The video signal is enabled only when DISPEN is high.
CURSOR	19	O	Cursor: Active high signal that is mixed with the video signal to produce a cursor display on the CRT screen.
CCLK*	21	I	Character clock: Input signal that defines character timing for CRTC display operation. Used to synchronize all CRT control signals on high to low transition.
XIOW*	22	I	Buffered I/O Write: High impedance input that controls the direction of data transfer between the CRTC internal register file and the 8088 CPU. Active high enables 8088 CPU read from CRTC. Active low enables CPU write into CRTC.
6845E	23	I	6845 (CRTC) enable: High impedance active high input that enables the data bus I/O buffers and clocks data into/ from the CRTC internal registers. Signal is derived from high to low transition of 8088 clock signal CLK coming in from expansion backplane bus.
BA0	24	I	Buffered address 0: Active high input that selects either address register or one of 18 data registers inside CRTC. Active low selects address register. Active high selects control data registers. Derived from lowest bit of 8088 CPU buffered address bus.
6845CS*	25	I	6845 (CRTC) chip select: Active low signal that enables read/ write operation to CRTC internal register file. Active when address from CPU is valid and stable.
BD0-BD7	26-33	I/O	Buffered data: Bidirectional bus used to transfer data between 6845 CRTC internal register file and 8088 CPU. Output drivers are tristate (high impedance) except during CPU read of CRTC.
RA0-RA3	35-38	O	Raster address: Four of five outputs from internal raster counter. Used to select the row (raster) of the character generator ROM.
HSYNC	39	O	Horizontal sync: Active high output that determines the horizontal position of displayed text. This signal directly drives a monitor, or it can be processed to generate composite video.
VSYNC	40	O	Vertical sync: Active high output that determines the vertical position of displayed text. With a pulse width fixed at 16H (horizontal raster lines), this signal can directly drive the monitor, or it can be processed to generate composite video.

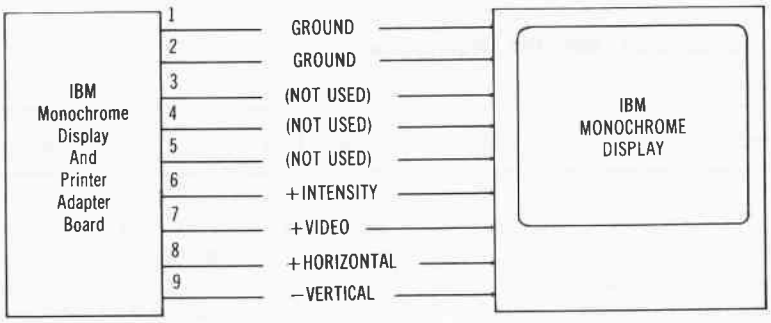


Fig. 2-64. The signals present on the pins of the connector of the monochrome adapter card.

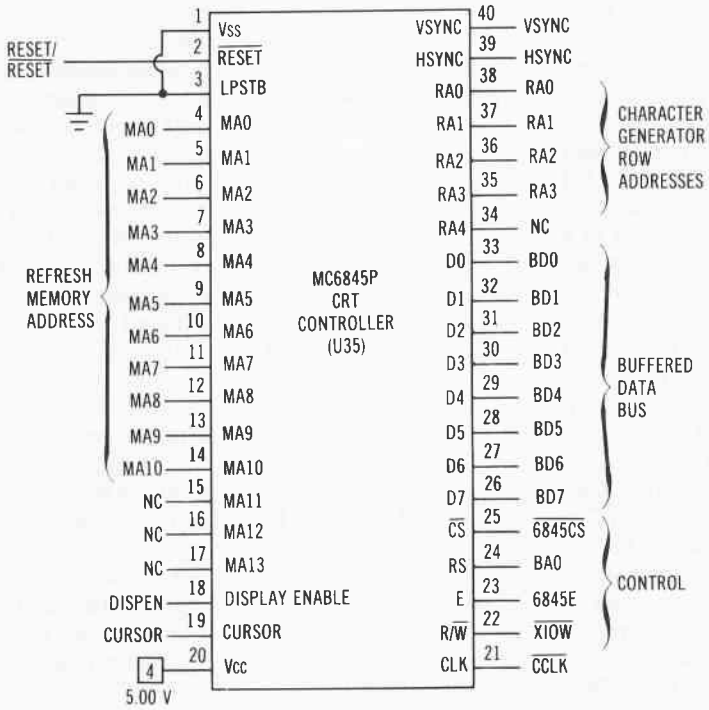


Fig. 2-65. 6845 CRT controller pin assignments (monochrome adapter board).

strobe input enables the chip to capture the refresh address in an internal light-pen register.

A programmable linear address register and 18 data registers are used in the CRTC to produce the necessary control signals for raster scan display generation. Table 2-21 describes the size and function of these registers.

Operation of the CRTC begins at system initialization when the system reset pulse clears all the counters and pulls the outputs low. As shown in Fig. 2-67 (CF page 2), the RESET signals are derived from the RESET DRV coming in on P3-B2 of the expansion bus backplane. The signal becomes RESET at the pin 3 output of 74LS125 quad 3-state buffer (U59) and is applied to the inputs of 74LS04 hex inverter (U56), 74LS393 dual binary ripple counter (U28), and 74LS86 quad 2-input exclusive OR gate (U54). Hex inverter U56 produces the active low RESET* signal that is applied to the rest of the circuitry shown in Fig. 2-67. Either U56 pin 12 or an optional RESET* signal from 74LS86 quad 2-input exclusive OR gate (U54) is connected to input pin 2 of 6845 CRTC (U35).

The active low RESET only functions when the LPSTB (pin 3) line is pulled low. Therefore, unused pin 3 is tied to ground on the monochrome monitor/printer adapter. One clock cycle after RESET occurs, the refresh memory address and row address bus outputs are pulled low. Immediately after the RESET signal is removed, the CRTC starts the display operation.

Table 2-21. Description of Internal Registers of 6845 CRTC

Reg.	Bits Used	R/W	Function
AR	5	W	Address register: A write-only pointer to one of 18 data registers (RA0-RA17). When 6845E and BA0 are both low, this register is addressed. When BA0 is high, the 18-register file is addressed.
R0	8	W	Horizontal total register: Write-only register that determines horizontal sync frequency. Used to program the total number of horizontal characters per line including the retrace period.
R1	8	W	Horizontal displayed register: Determines number of characters that will be displayed per line.
R2	8	W	Horizontal sync position register: Determines horizontal sync positions on horizontal line as multiple of character clock period.
R3	4	W	Horizontal sync width register: Determines width of horizontal sync pulse. Programmed as multiples of character clock period to allow compensation for different horizontal sync widths on certain monitors.
R4	7	W	Vertical total register: Determines the total number of lines per frame including vertical retrace period.
R5	5	W	Vertical total adjust register: To adjust total number of rasters per field and thus vertical deflection frequency.
R6	7	W	Vertical displayed register: Determines number of displayed character rows on CRT screen. Programmed in character row times.
R7	7	W	Vertical sync position register: Determines vertical sync screen position as multiple of horizontal character line periods. Increasing value shifts display position up. Decreasing value shifts display position down.
R8	2	W	Interlace mode register: Controls raster scan mode. 0 0 = noninterlace mode 0 1 = interlace sync mode 1 0 = noninterlace mode 1 1 = interlace sync & video mode In noninterlace mode used on IBM PC, the rasters of even and odd number fields are duplicated.
R9	5	W	Maximum scan line address register: Determines number of raster scan lines per character row including spacing.
R10	7	W	Cursor start register: Lower 5 bits determine cursor start raster address. Upper 2 bits define cursor display mode with top bit for blink and next bit for the period. B P Cursor Display Mode -- -- ---------- 0 0 Nonblink 0 1 Cursor nondisplay 1 0 Blink period 16X field period (blink frequency 1/16 of vertical field rate) 1 1 Blink period 32X field period
R11	5	W	Cursor end register: Sets end raster address (see Fig. 2-66).
R12	6	W	Start address register (high): Combines with R13 to define first address generated as refresh address after vertical blanking.
R13	8	W	Start address register (low): Lower 8 bits of the 14 bit refresh address.
R14	6	R/W	Cursor register (high): Combines with R15 to define cursor location. Top 2 bits always zero.
R15	8	R/W	Cursor register (low): Lower 8 bits of 14 bit screen location of cursor.
R16	6	R	Light pen register (high): Combines with R17 to capture and store the detection address of the light pen. Because of slow light pen response, the value must be corrected (calibrated) by the BIOS program. Contents of internal address counter strobed into R16 and R17 on following high to low transition of CCLK after LPSTB goes high. LPSTB tied low on monochrome display/printer adapter so cannot use light pen with this board.
R17	8	R	Light pen register (low): Lower 8 bits of 14-bit detection address defined in description for R16.

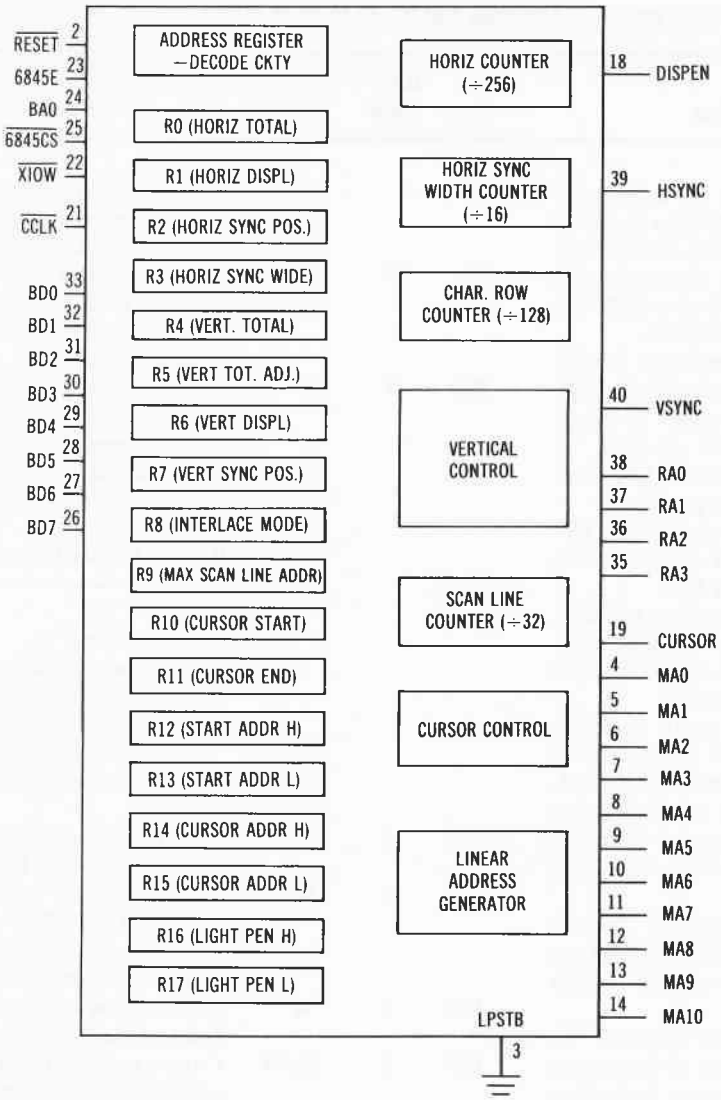


Fig. 2-66. Block diagram of 6845 CRT controller.

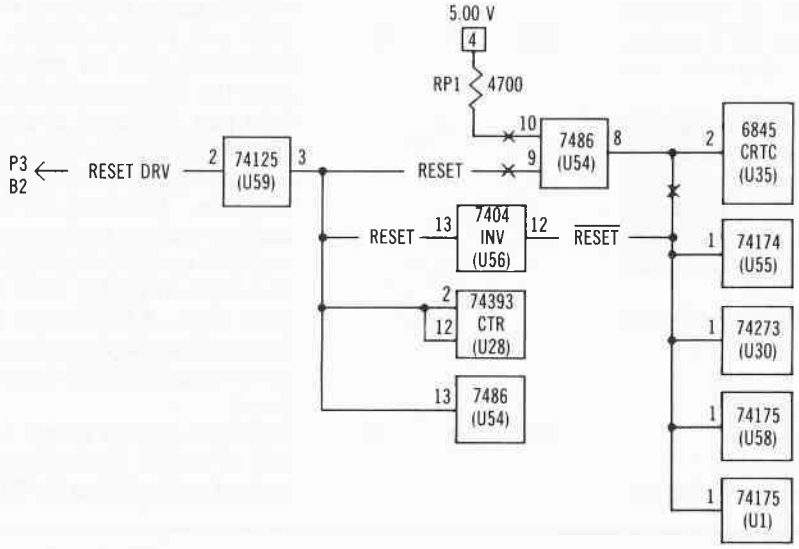


Fig. 2-67. Monochrome adapter board reset circuitry.

The same system reset that generated the RESET DRV signal passed out on the expansion bus also started the various clock signals to begin. The internal operation of the CRTC is synchronized by the control character clock input (CCLK) on pin 21 (Fig. 2-65). This signal enables the 8088 CPU to communicate with the 6845 CRTC over a buffered 8-bit data bus.

Control clock signal CCLK and its alternate CCLK* are generated by the circuitry shown in Fig. 2-68. Either the 16.257 MHz oscillator in the lower left of the figure or a TEST OSC signal are multiplexed through 74LS153 (U24) to produce DOTCLK. This signal is used to clock the 74LS112 Dual J-K negative edge-triggered flip-flop (U5). The delayed output from 74LS174 hex D flip-flop U1 is applied directly to the pin 3 J input to U5 and via the 74LS04 inverter (U2) to the pin 2 K input. The Q output from U5 is NORed with the delayed (Q5) signal from U1 to produce an active low CCLK* output on pin 13 of U57. CCLK* is applied to the pin 21 clock input to 6845 CRTC U35.

CCLK* is used to clear the 74LS74 dual D flip-flop U4 and is inverted by 74LS04 hex inverter U2 to produce CCLK. CCLK is ANDed with the outputs from U4 pin 6 and U25 pin 11 to generate a control address chip select control clock signal CACS CCLK on output pin 6 of U26.

With the adapter board clock signals pulsing and the 4.772727 MHz CLK signal coming onto the adapter board from the expansion bus, the monochrome card begins to stir. As shown in Fig. 2-69, the active low buffered I/O read and write inputs to 74LS00 quad 2-input NAND gate (U53) produce an active high enable signal E that is passed out onto the board and into the D (pin 12) and clear (pin 13) inputs to 74LS74 dual D flip-flop U45. Pin 13 needs an active low signal to clear the flip-flop, therefore, on the rising edge of the next CLK input on pin 11, the Q output (pin 9) toggles producing the 6845 CRTC enable signal 6845E that is applied to pin 23 of the CRTC U35.

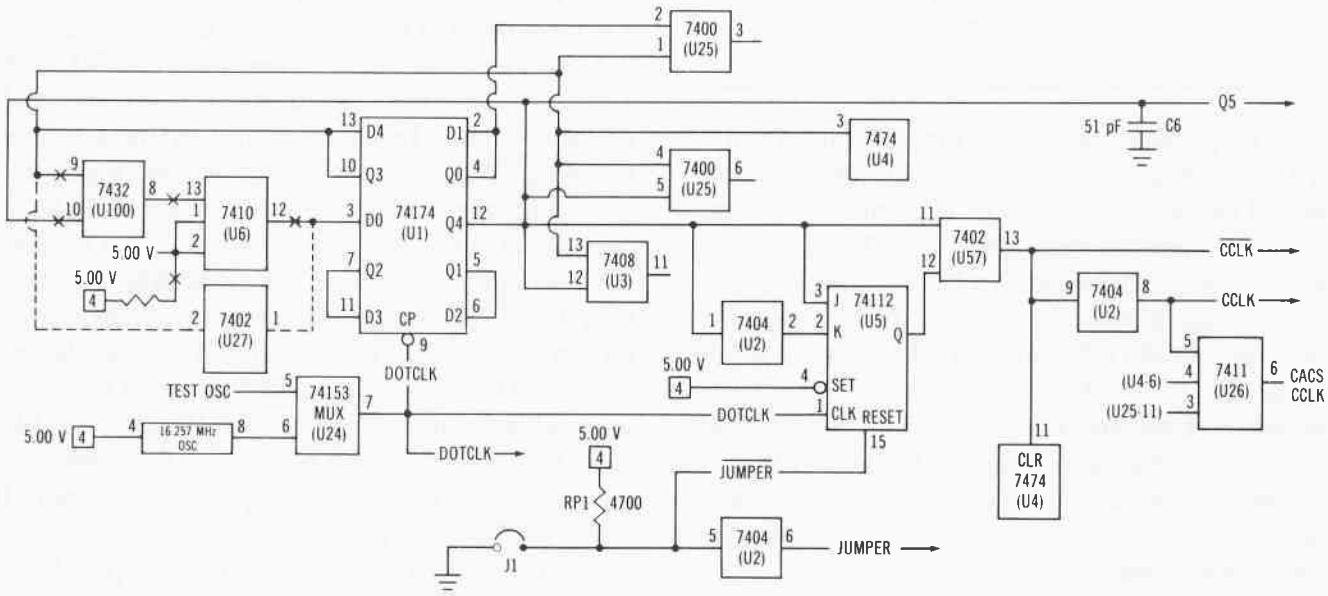


Fig. 2-68. 6845 clock (CCLK*) circuitry.

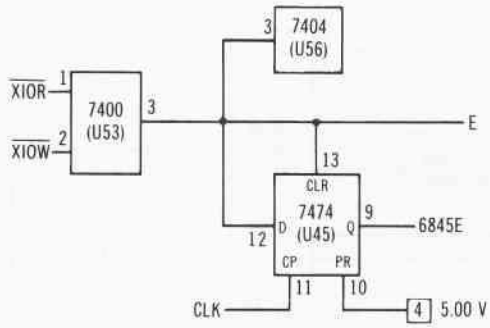


Fig. 2-69. 6845 chip enable signal (6845E).

With the CRTC enabled, the address bus and the system board address enable signal AEN are passed onto the adapter board and decoded by 74LS138 1-of-8 decoder/demultiplexer U52 as shown in Fig. 2-70.

The adapter board is addressed as an I/O device using hex addresses 3B4H, 3B5H, 3B8H, 3BAH, 3BCH, 3BDH, 3BE, and 3D8 through 3D1, as shown in Table 2-22.

Table 2-22. I/O Monochrome Adapter Board Addressing

I/O Address	Accesses
3B4	6845 index (address) register
3B5	6845 data registers
3B8	CRT control port 1
3BA	CRT status port
3BC	Parallel data port
3BD	Printer status port
3BE	Printer control port

To generate an active low output on pin 7 of U52 in Fig. 2-70, an address of 3Bx must be placed on the address bus. The suffix "x" must be a hex value less than eight. The 3B part of the address produces an active low output from pin 7. The signal becomes a qualifying input for another 74138 (U50) and 74LS32 quad 2-input OR gate U43. Buffered address signal BA3 is generated from address bit A3. The ORed output is the 6845 chip select signal (6845CS*) that is applied to pin 25 of CRTC U35. Thus anytime 3B4 or 3B5 are placed on the address bus, the CRTC is accessed.

The active low signal 6845CS* is inverted by U44 to produce the active high 6845CS signal that is used elsewhere on the adapter board.

The board has been reset, the clocks started, and the CRTC enabled and selected. The video system wakes up and the waveforms described in Fig. 2-71 are produced.

Mode Control Port 1 (3B8H)

The first command sent to the board by the 8088 CPU is an address to I/O location 3B8H (CRT control port 1). The value placed on the system address bus is 01H. This value sets the monochrome board to the high-resolution mode. This is a necessary prerequisite to accessing the monochrome adapter. The CRT mode control port 1 is formed by the circuitry shown in Fig. 2-72 (See CF (CSCS2-A) pages 2, 15, and 16).

The port has the I/O address 3B8H (0 0 1 1 1 0 1 1 1 0 0 0 in binary). With this code, bits 3, 4, 5, 7, 8, and 9 are high. Therefore, the 74LS138 1-of-8 decoder U52 inputs A4, A5, A7, A8, and A9 are high. Inputs A6 and AEN are low enabling U52 to generate an active low signal out pin 7 into the enable input of another 74LS138 decoder (U50). BA0, BA1, and BA2 are all low. BA3 is high. With either XIOR or XIOW active low, 74LS00 quad 2-input NAND produces a high on pin 3 that is felt on input pin 3 to 74LS04 hex inverter U56. The low on output pin 4 becomes an enable input to U50 completing the enable code (pin 5 low and pin 6 high are the other two enable inputs) and generating a low select port 1 signal (SEL1*) out pin 15.

SEL1* is the clock input on pin 9 of 74LS175 quad D flip-flop U58. A hex 01 is on the data bus input so BD0 is 1, BD2, BD3, and BD5 are all low (binary 0). When SEL1* goes

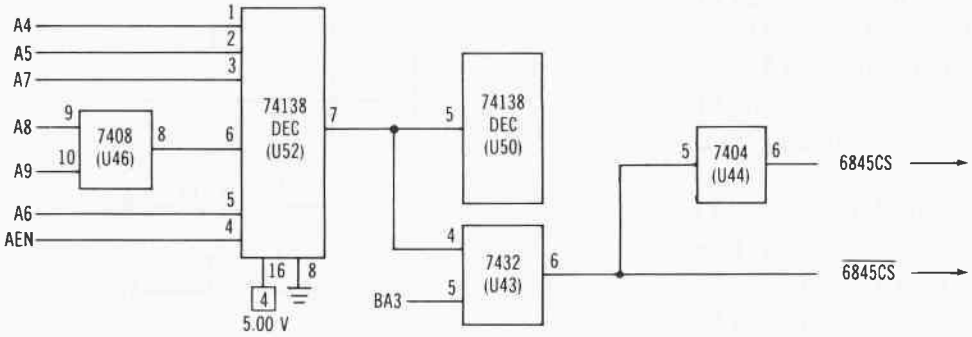


Fig. 2-70. 6845 chip select circuitry.

low to high, the high on BD0 is loaded in pin 4 generating a low out the Q* output pin 3. This low signal is felt on pin 4 of 74LS00 2-input NAND U53 producing a high on output pin 6. The high out pin 6 is felt on input pin 11 to 74LS10 3-input NAND U6. The other two inputs to U6 are held high so the pin 8 output becomes the active low high resolution signal HRES*.

By changing the input data word (BD0, BD2, BD3, and BD5) on the inputs to U58, other modes of operation can be selected. Table 2-23 describes the various modes that can be produced.

Status Port (3BAH)

The 8088 CPU (U3 on system board) can monitor the status of the video circuitry by addressing hex 3BA (0 0 1 1 1 0 1 1 1 0 1 0 in binary). The lower four bits of the address (BA0, BA1, BA2, and BA3) on the input to U50 in Fig. 2-72 produces an active low output on pin 13 (STATUS SEL*). This signal is passed to the enable input (pin 1) of 74LS244 octal buffer (U60) shown in the upper right of Fig. 2-73.

When enabled, U60 passes the status of the horizontal drive and line video signals onto

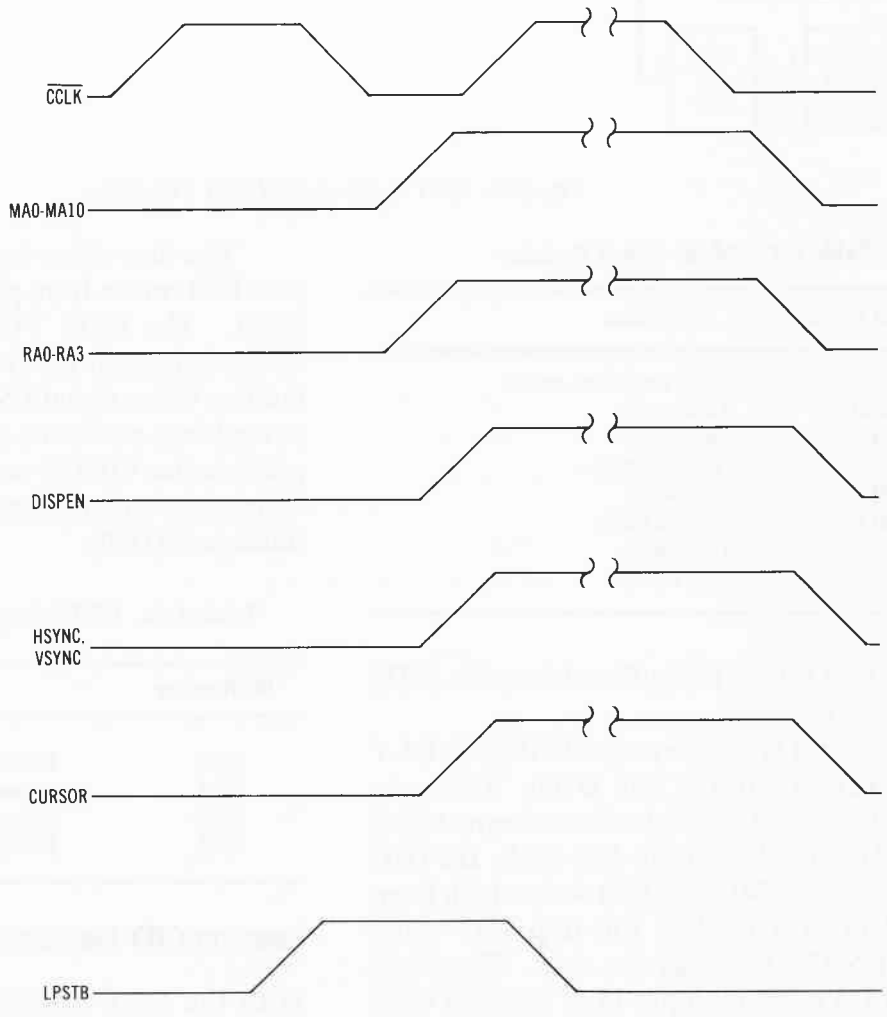


Fig. 2-71. 6845 CRT controller timing diagram.

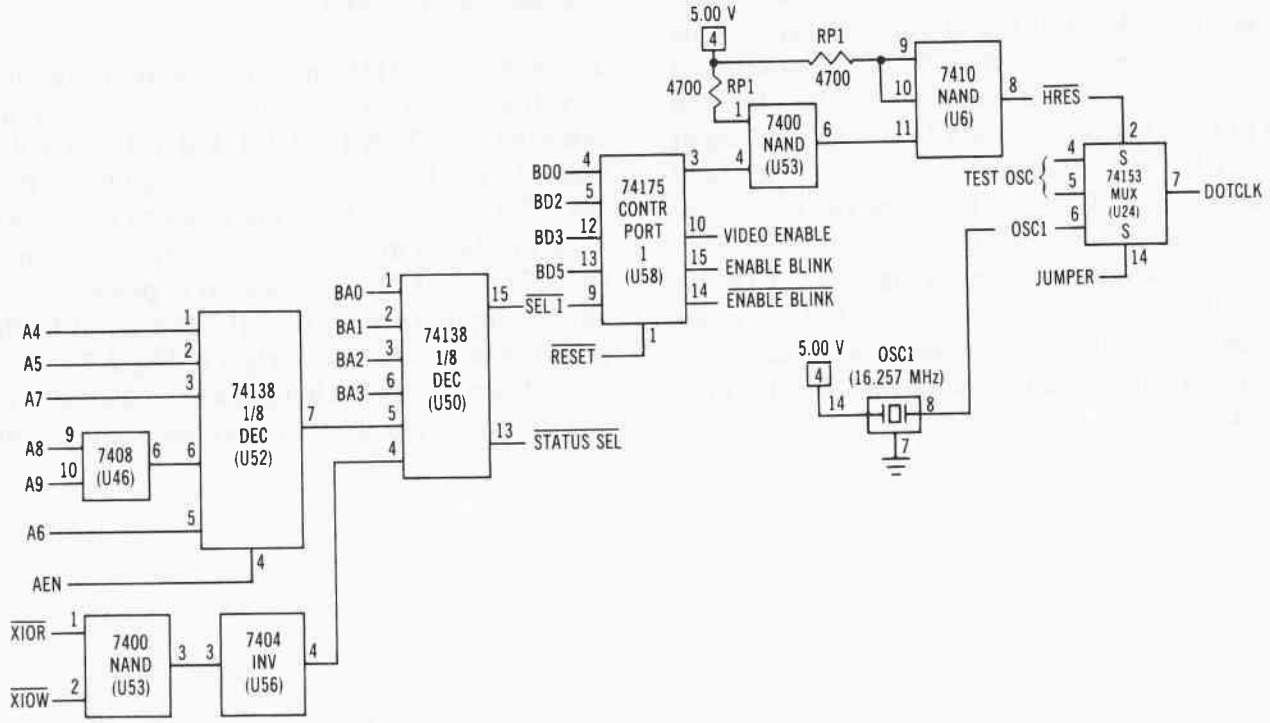


Fig. 2-72. CRT mode control port 1 circuitry.

Table 2-23. Mode Select Register

Data Bit Number	Function
BD0	High resolution mode
BD1	(not used)
BD2	(not used)
BD3	Video enable
BD4	(not used)
BD5	Enable blink
BD6	(not used)
BD7	(not used)

the lower four bits of the buffered data bus (BD0 through BD3).

Horizontal sync delay signal HSYNC DLY is derived from 74LS32 NOR U100. This gate monitors the status of the horizontal sync signal input to 74LS174 hex D flip-flop U55. HSYNC DLY is ANDed in 74LS08 (U3) with a high from pin 5 of 74LS74 dual D flip-flop U45 when VIDEO ENABLE clocks the chip. The pin 6 output of U3 is input on pin 13 of 74LS244 octal output buffer U64 to generate the horizontal drive signal (HORIZ DRIVE) out pin 7.

The line video input to pin 8 of the status port U60 comes from pin 5 of 74LS74 D flip-flop U101. The B/W VIDEO signal on pin 2 is clocked through U101 by DOTCLK to become the line video signal LVIDEO. LVIDEO is also passed into pin 15 of the output buffer U64 to produce the VIDEO output on pin 5. Table 2-24 defines the buffered data bus outputs of the CRT status port (U60).

Table 2-24. CRTC Status Port Bit Assignments

Bit Number	Function
BD0	Horizontal drive status
BD1	(reserved)
BD2	(reserved)
BD3	Black/white video status

Loading CRT Registers

With the CRT controller (U35) selected and enabled, the 8088 CPU performs an OUT instruction to I/O location 3B4. This addresses

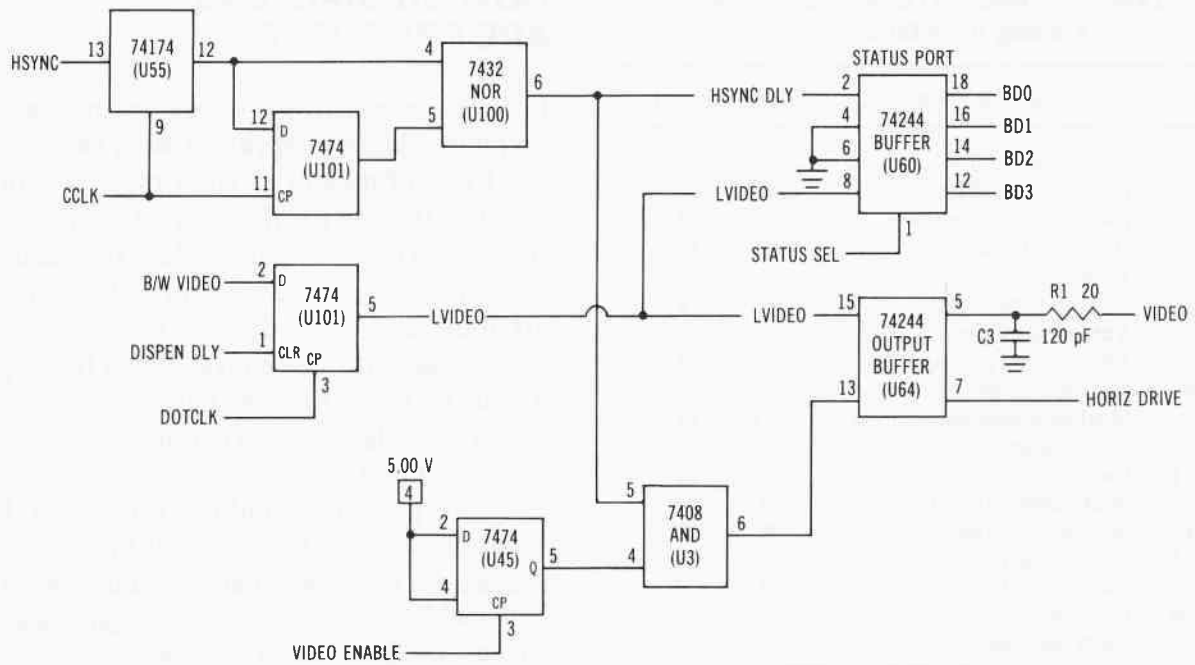


Fig. 2-73. CRTC status port circuitry.

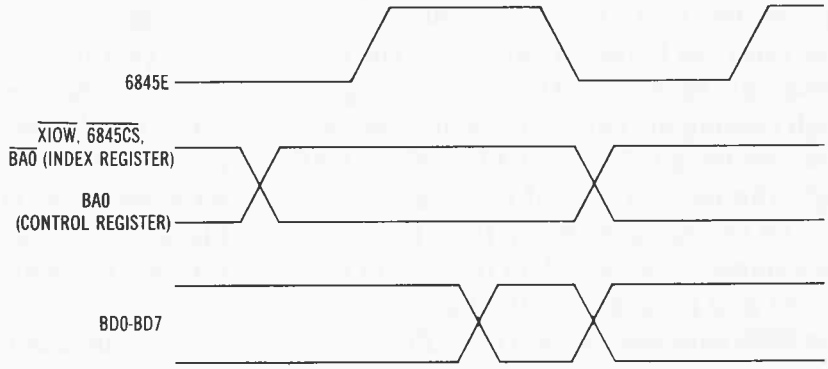


Fig. 2-74. Writing information into the 6845 CRT controller via the buffered data bus.

the 6845 index register. The lower 5 bits on the CPU data bus is a code representing 1 of the 18 control data registers. The code locks the CRTC into accessing a particular data register when the 8088 performs its next instruction, an OUT to location 3B5. The inversion of the least significant bit (BA0) selects the 18 control register stack. Because only 1 of the 18 registers is enabled, the information on the buffered data bus is copied directly into the selected and enabled control register. Figure 2-74 shows the timing relationships for writing information into the 6845 CRT controller via the buffered address bus.

Signal 6845E, chip enable goes high, XIOW* goes low, 6845CS* goes low, and BA0 goes low selecting the index address register. Data on the buffered data bus (BD0 through BD7) represents one of the 18 control registers. Then, with 6845E still high, XIOW* still low, and 6845CS* low, new data is placed on the buffered data bus and BA0 shifts high latching the new data into the selected control register.

Table 2-25 summarizes the 6845 CRTC internal register utilization and access addressing for use on the IBM monochrome display/printer adapter.

Table 2-25. 6845 CRTC Internal Register Loading upon Initialization

Register	Function Value	Hex	Decimal
R0	Horizontal total	61	97
R1	Horizontal displayed	50	80
R2	Horizontal sync position	52	82
R3	Horizontal sync width	0F	15
R4	Vertical total	19	25
R5	Vertical total adjust	06	6
R6	Vertical displayed	19	25
R7	Vertical sync position	19	25
R8	Interlace mode	02	2
R9	Maximum scan line address	0D	13
R10	Cursor start	0B	11
R11	Cursor end	0C	12
R12	Start address (high)	00	0
R13	Start address (low)	00	0
R14	Cursor (high)	00	0
R15	Cursor (low)	00	0
R16	Light pen (high)	-	-
R17	Light pen (low)	-	-

Reading Information Out of CRTC

Figure 2-75 describes the timing relationships for reading information from the 6845 CRT controller onto the buffered data bus. The chip is enabled and selected. Here, XIOW* goes active high causing the chip to recognize the data flow direction from CRTC to CPU. When BA0 goes high, the data in one of the four control registers (R14 through R17) that has been previously selected, is placed on the output data pins of U35 and passed over the buffered data bus to the 8088 data bus and into the CPU.

HOW CHARACTERS ARE PRODUCED

Both the monochrome display/printer and color graphics adapters represent each character with two bytes of memory. The first byte represents the ASCII code for the character. (A listing of the ASCII character codes is included in Appendix G.) The second byte contains an attribute code that defines the video features associated with the character to be displayed. These features include background darkness, foreground darkness, brightness (intensity), and blink condition.

Six possible combinations exist for the attribute byte. One, normal video with light characters on a dark background; two, reverse video with dark characters on a light background; three, flashing light characters on a dark background; four, flashing dark characters on a light background; five, invisible light characters on a light background; and six, invisible dark characters on a dark background. The last two combinations are of little use.

Two interlaced memories are installed on the monochrome adapter board. The character codes for the characters which are produced on the monitor screen are stored in 2K bytes of static memory (U12, U13, and U14), as shown in Fig. 2-76. U13 and U14 contain character codes for the top half of the display screen. The character codes for the bottom half of the screen are contained in U12 and U14.

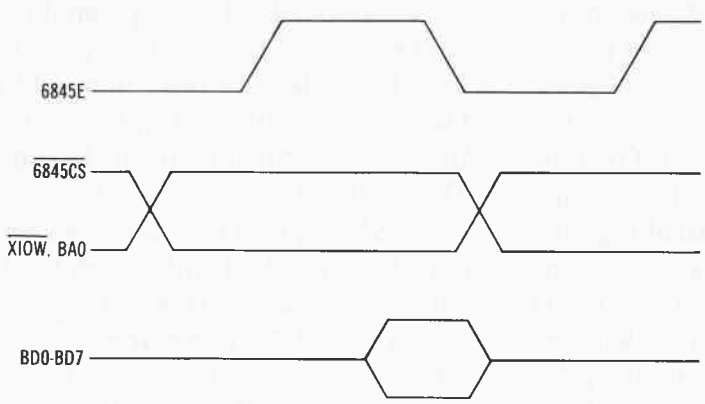


Fig. 2-75. Reading information from the 6845 CRT controller onto the buffered data bus.

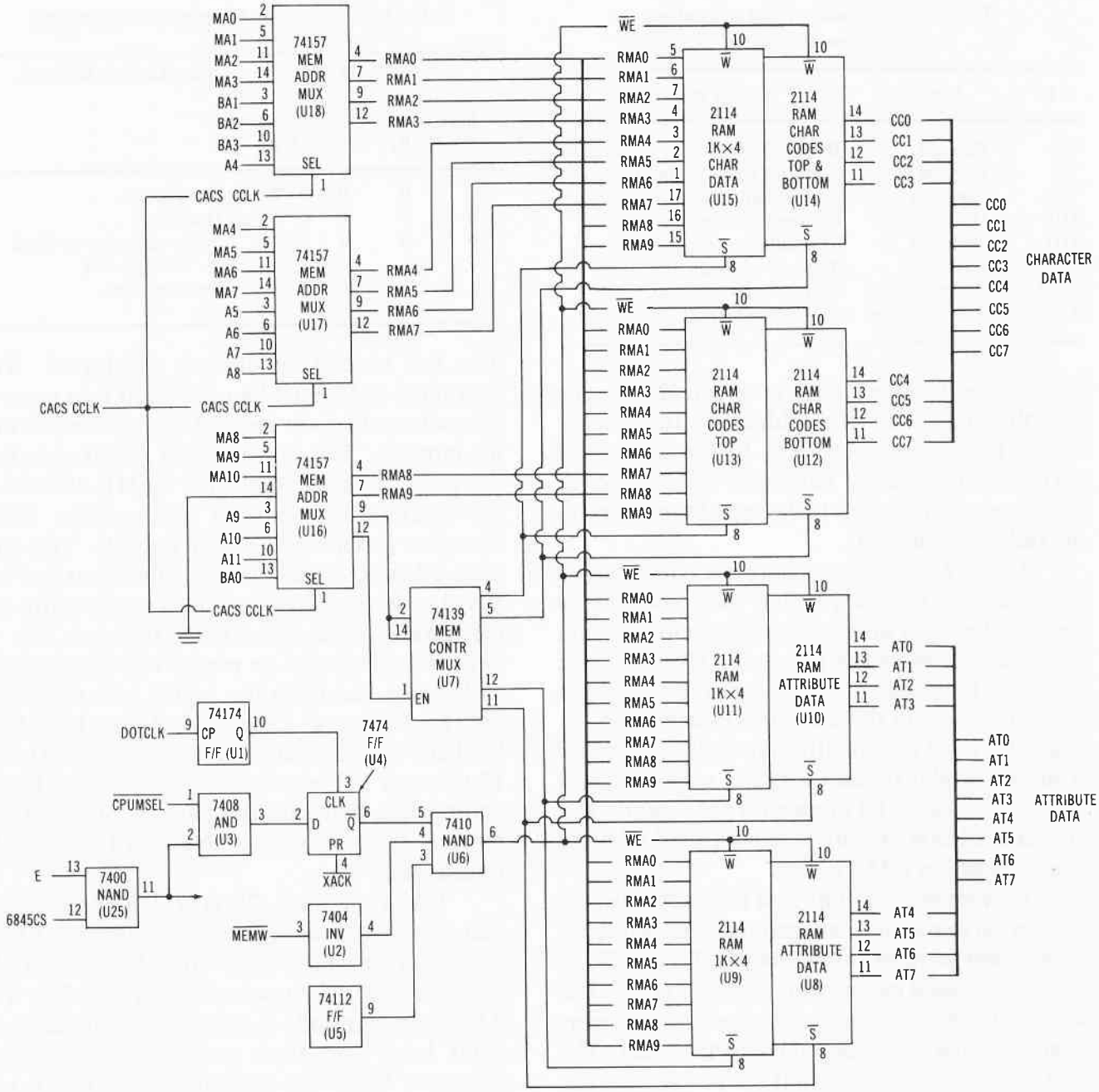


Fig. 2-76. Display buffer memory.

The blink, highlight, reverse video, and underline attribute features are stored in another 2K bytes of static RAM (U8, U9, U10, and U11). Table 2-26 describes the memory allocation for monochrome display video attribute information. The blink feature for the top half of the screen (first 1024 characters) is stored in U8. The highlight and underline features for the top half of the screen are stored in U11. These features for the bottom half of the screen are stored in U10. Reverse video information for the top half of the display screen is stored in U9 and U11. Reverse video for the bottom half of the screen is stored in U8 and U10.

Video information from the system board is stored in a fast static 4K RAM. Containing combined character codes and attribute codes, the 4K static RAM display buffer holds the video

Table 2-26. Memory Chip Allocation for Video Attributes

IC	Attribute	Screen Pixel Area
U8	Blinking	Bottom 976 character positions
U9	Blinking	Top 1024 character positions
U10	Highlight	Bottom half of screen
U10	Underline	Bottom half of screen
U11	Highlight	Top half of screen
U11	Underline	Top half of screen
U8, U10	Reverse video	Bottom half of screen
U9, 11	Reverse video	Top half of screen

Table 2-27. Attribute Feature Enhancements

Attribute Bits						Video Feature Achieved
Background			*Foreground*			
B6	*B5*	*B4*	*B2*	*B1*	*B0*	
0	0	0	0	0	0	Nondisplay
0	0	0	0	0	1	Underline
0	0	0	1	1	1	White character on black background
1	1	1	0	0	0	Reverse video

information for one complete screen (25 rows by 80 columns). The first address in the display buffer (B0000) corresponds to the upper-left corner of the screen. The lower-right corner of the screen corresponds to the top of the 4K memory (address B0F9FH).

Up to 256 different character codes can be stored in the 4K display buffer. Each character is represented by a unique character and attribute code pair, as shown in Fig. 2-77. Even address locations hold the character codes; odd address locations contain the attribute information. A character code and an attribute code are fetched from the display buffer every 552 ns producing a video data rate of 1.8 megabytes per second. The contents of this 4K buffer can be read into the system board by DMA action.

Combining the blink and intensity bits with the foreground and background bits produces other video features as described in Table 2-27.

The second memory on the adapter board is an 8K ROM that contains the dot patterns (fonts) for the 256 different character codes. The ASCII character code that enters the monochrome adapter board from the system board data bus cannot be directly displayed. The character code must be converted into rows of dot patterns that are sent out the video cable into the monitor. The conversion of ASCII code into dot patterns is accomplished by MK-36906N-4, an MK36000-compatible 8K-by-6-bit ROM character generator U33 (CF page 2). This chip is an edge-activated NMOS ROM that has on-board address latches controlled by the active low chip enable input signal CEROM* (pin 20). A negative-going edge on pin 20 will activate U33 and strobe and latch the inputs into the chip's address registers. Once the address hold time has been met, the outputs (pins 9 through 11, and 13 through 17) become active and contain the appropriate character dot pattern. The outputs remain latched and active until CEROM* returns high.

When the 6845 CRTC (U35) determines that display data is required, it causes the ASCII code and attribute data stored in the RAM display buffer to be read out (see Fig. 2-78). The 8-bit character code is temporarily stored in an octal latch and then passed into character generator U33. The other input into U33 is a 4-bit address defining which row of dots for the

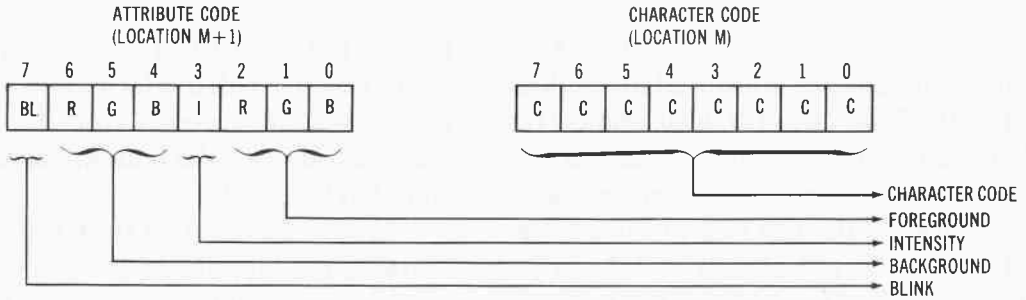


Fig. 2-77. Character and attribute codes.

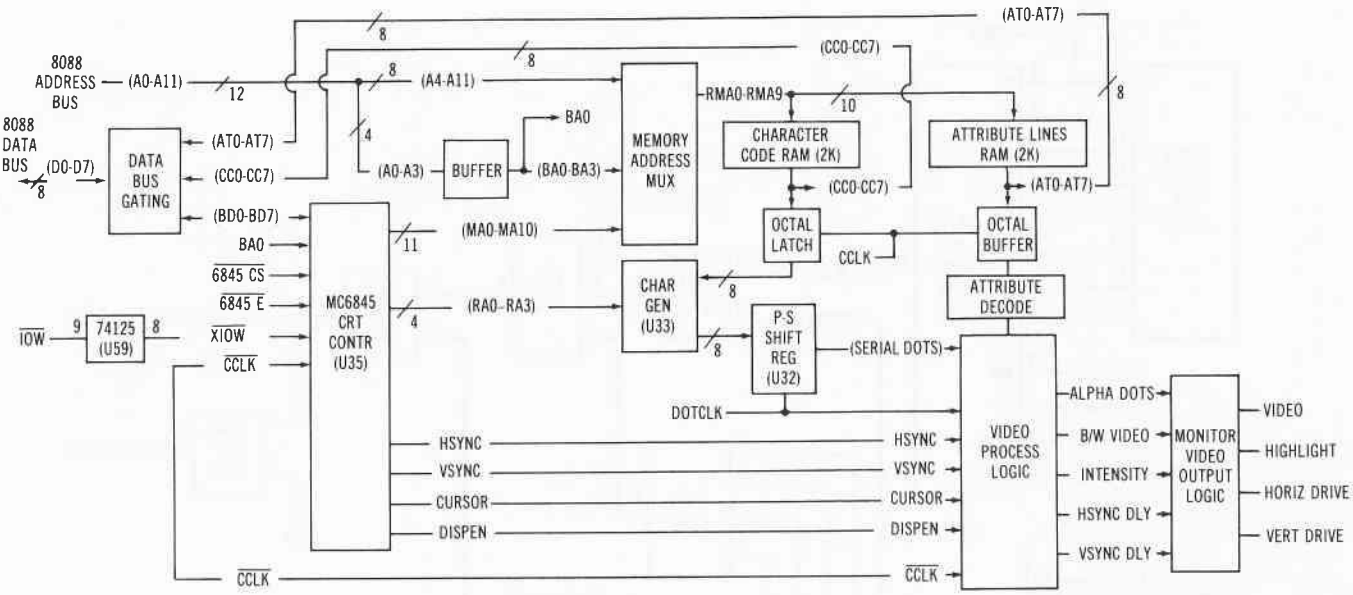


Fig. 2-78. Monochrome display adapter block diagram.

character to be displayed will be passed out U33 into a 74LS166 parallel-to-serial shift register (U32) to be converted into a serial dot stream at its output.

The attribute information is applied to video process logic with the serial dot patterns to produce a modified dot pattern with the desired attribute features. This combined signal is clocked by CCLK* through the logic to become ALPHA DOTS, a select signal to enable a blink/no-blink intensity attribute signal, and a black and white video serial dot stream (B/W VIDEO). B/W VIDEO passes through a monitor direct drive output buffer to become the VIDEO signal at the unit video connector J3. Figure 2-79 shows the circuitry that produces the ALPHA DOTS and B/W VIDEO signals that are applied to the monitor direct drive output logic (see also CF page 2). An 8-bit code is output by character generator U33 onto its character generator bus (CGB0 through CGB7). This code enters the data input buffer of 74LS166 parallel-to-serial shift register U32 on pins 2 through 5, 10 through 12, and 14.

An active low clear video (CLRVIDEO*) signal is used as a master reset input on pin 9 of U32. The chip is clocked by the DOTCLK input on pin 7. Pin 15 is for the parallel enable serial load signal S/L*. When S/L* is low one setup time before the low-high clock transition, parallel data from U33 enters U32. This same signal prevents a serial input from entering the chip via SERIN on pin 1. Once the character generator bus has been entered into the eight internal R-S flip-flop latches inside U32, SERIN is held low and S/L* is pulled high letting DOTCLK shift the 8-bit dot pattern code from least significant bit to most significant bit out the Q7 register output (pin 13) as a sequence of serial dots (S DOTS). The character generator produces a row of 8 dots. The character box is 9 dots wide so a blank dot is inserted between each character. SERIN is used to duplicate the eighth dot into the ninth dot position for ASCII characters whose codes are between B0H and DFH.

The S DOTS bit stream is ORed with the UNDERLINE signal on pin 13 of 74LS32 quad 2-input OR U43 to produce a modified bit stream input to pin 11 of 74LS11 3-input AND gate U26. The other two inputs to U26 are NODSPLY* from 74LS175 quad D flip-flop U29 and attribute control from the pin 6 output from 74LS20 dual 4-input NAND U62. The pin 8 output from U26 is ANDed with the combined CURSOR DLY and CURSOR BLINK signals from 74LS08 quad 2-input AND U3 in another part of 74LS32 (U43) to produce the signal ALPHA DOTS. These ALPHA DOTS becomes one input to 74LS86

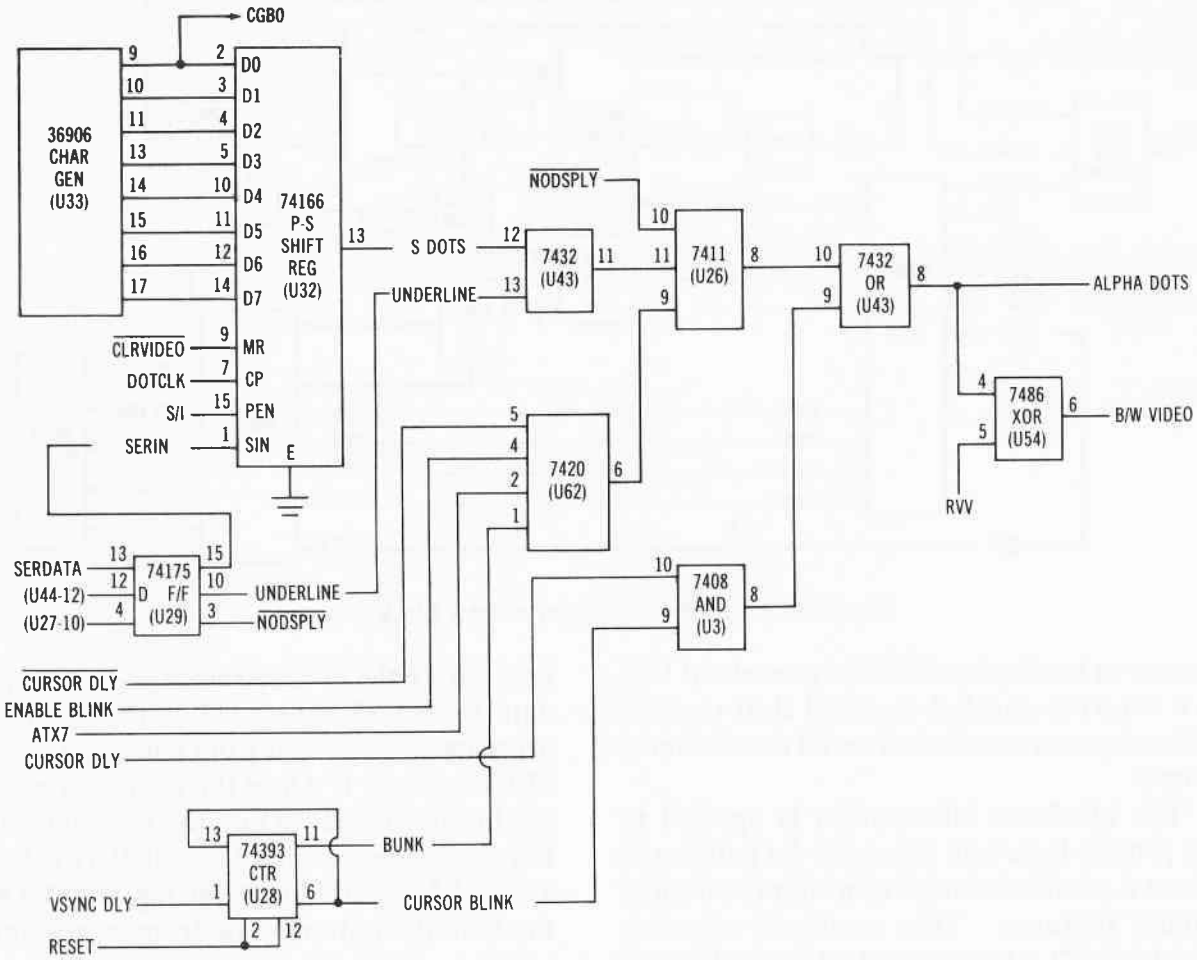


Fig. 2-79. Video display circuitry.

quad 2-input exclusive OR gate U54. A reverse video signal RVV on pin 5 is the other input. The EXOR output on pin 6 becomes the black and white video signal B/W VIDEO that reaches the J3 connector as the signal VIDEO.

The horizontal and vertical sync signals (HSYNC, VSYNC), CURSOR, and DISPEN signals are also applied to the video process logic. HSYNC and VSYNC become HSYNC DLY and VSYNC DLY inside the logic and end up as horizontal and vertical drive signals (HORIZ DRIVE, VERT DRIVE) at the video output connector J3, as shown in Fig. 2-80.

The pins 39 and 40 output from 6845 CRT controller U35 become inputs to 74LS174 hex D flip-flop U55. HSYNC on D input pin 13 is passed out the Q side of the latch on pin 12. This signal is delay-latched through 74LS74 U101 and then ANDed in 74LS32 U100 to produce HSYNC DLY on output pin 6. A logic high is clocked through 74LS74 U45 to appear at pin 4 of 74LS08 AND gate U3 when VIDEO ENABLE is active high on the pin 3 clock input of U45. The ANDed output from U3 is buffered through 74LS244 tristate octal output buffer U64 to become a 15,750 Hz stream of active low 5 microsecond horizontal drive (HORIZ DRIVE) pulses at video connector J3 pin 8.

The VSYNC output from pin 40 of 6845 CRTC U35 is latched through U55 to become VSYNC DLY on pin 1 of 74LS86 exclusive OR gate U54. Its other input (JUMPER*) comes from the J1, resistor, 74LS04 inverter U2 network. The output from U54 is buffered through U64 to become a 60 Hz stream of 190 microsecond pulses (VERT DRIVE) on pin 9 of connector J3.

The video monitor direct drive output logic

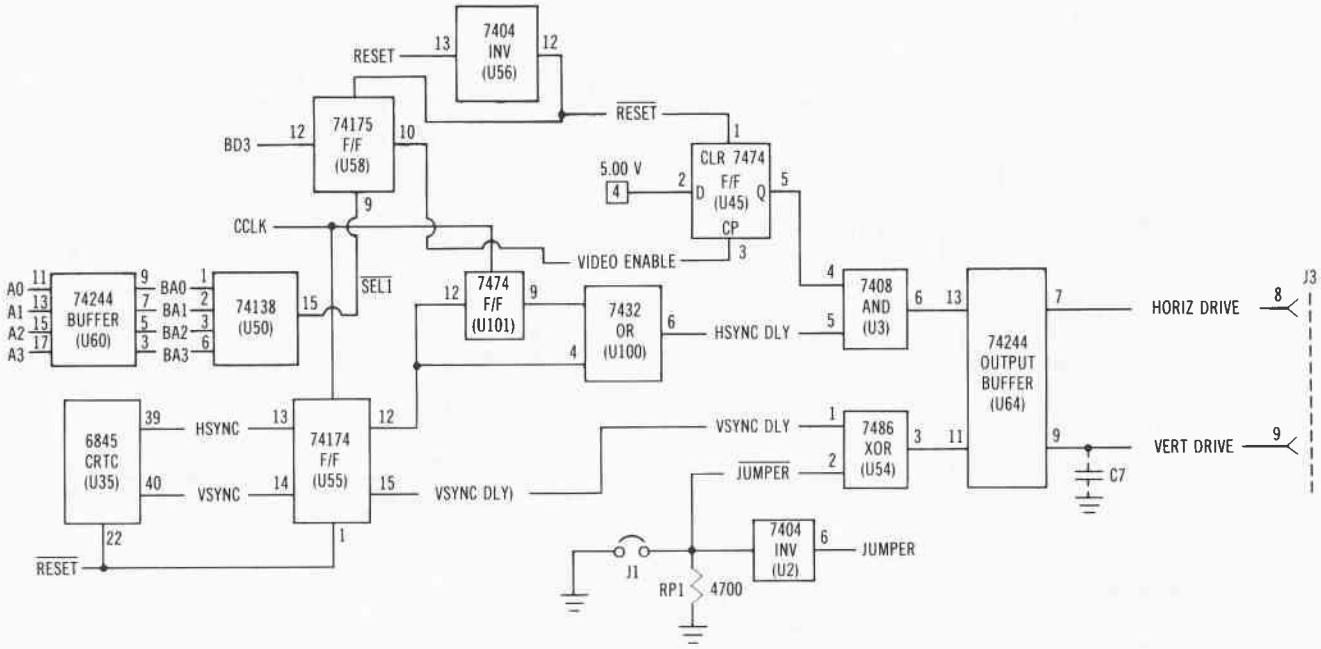


Fig. 2-80. Monochrome adapter horizontal and vertical sync circuitry.

is shown in Fig. 2-81. B/W VIDEO is clocked through U101 and U64 to become the VIDEO dot pattern information passed out J3 pin 7. HSYNC DLY is ANDed in 74LS08 U3 and passed through U64 to become HORIZ DRIVE pulses on J3 pin 8, and VSYNC DLY is exclusive ORed in 74LS86 U54 and buffered through U64 to become VERT DRIVE pulses on J3 pin 9.

The fourth important output from this logic circuitry is the intensity signal (HIGH LIGHT) on J3 pin 6. Figure 2-81 shows that this signal originates from the high bit (AT7X) of the attribute byte coming out of the 2K attribute RAM. As described with Fig. 2-77, the high bit can be toggled to turn on and off the blink function. Bit 3 (AT3X) of the attribute word is used to control the intensity of the character being displayed. Buffered attribute bits AT3X and AT7X are latched into two flip-flops in 74LS273 octal D flip-flop U30. The Q7 output (pin 19) is ANDed with (ENABLE BLINK)* to produce a bright blinking character control intensity blink signal I(B) on pin 3 of U46. The Q3 output (pin 9) is the full intensity control signal I(F).

Both intensity signals–I(B) and I(F) are applied to 74LS157 multiplexer U63. The pin 1 (ALPHA DOTS) and pin 15 (DISPEN DLY)* signals are used to select I(B) or I(F) as the pin 12 output. An active low DISPEN DLY signal on pin 15 enables ALPHA DOTS on pin 1 to select one of the two intensity input signals. When pin 1 is high, I(F) is selected and its condition is passed through to output pin 12. A low on pin 1 selects I(B) pin 14 as the signal to pass through to output pin 12.

The multiplexed output from U63 is buffered by U64 and becomes the signal HIGH LIGHT that is passed out through connector J3 pin 6. Figure 2-64 described the monochrome video adapter outputs at connector (J3).

So we have a serial stream of video dot information, clocking out over the video connector into the monochrome display. Figure 2-82 shows that the internal registers of the 6845 CRTC determine how many characters are displayed on the screen (R1=80), the total character dot arrays provided (R0=97), the number of raster lines for each character (R9=14), the number of vertical characters displayed (R6=25), and the vertical total offset (R5=6).

If we zoom in on a single character position (for example, C1 in Fig. 2-82), a 9 by 14 dot array can be noticed, as shown in Fig. 2-83A. The serial VIDEO stream of dot pulses coming out

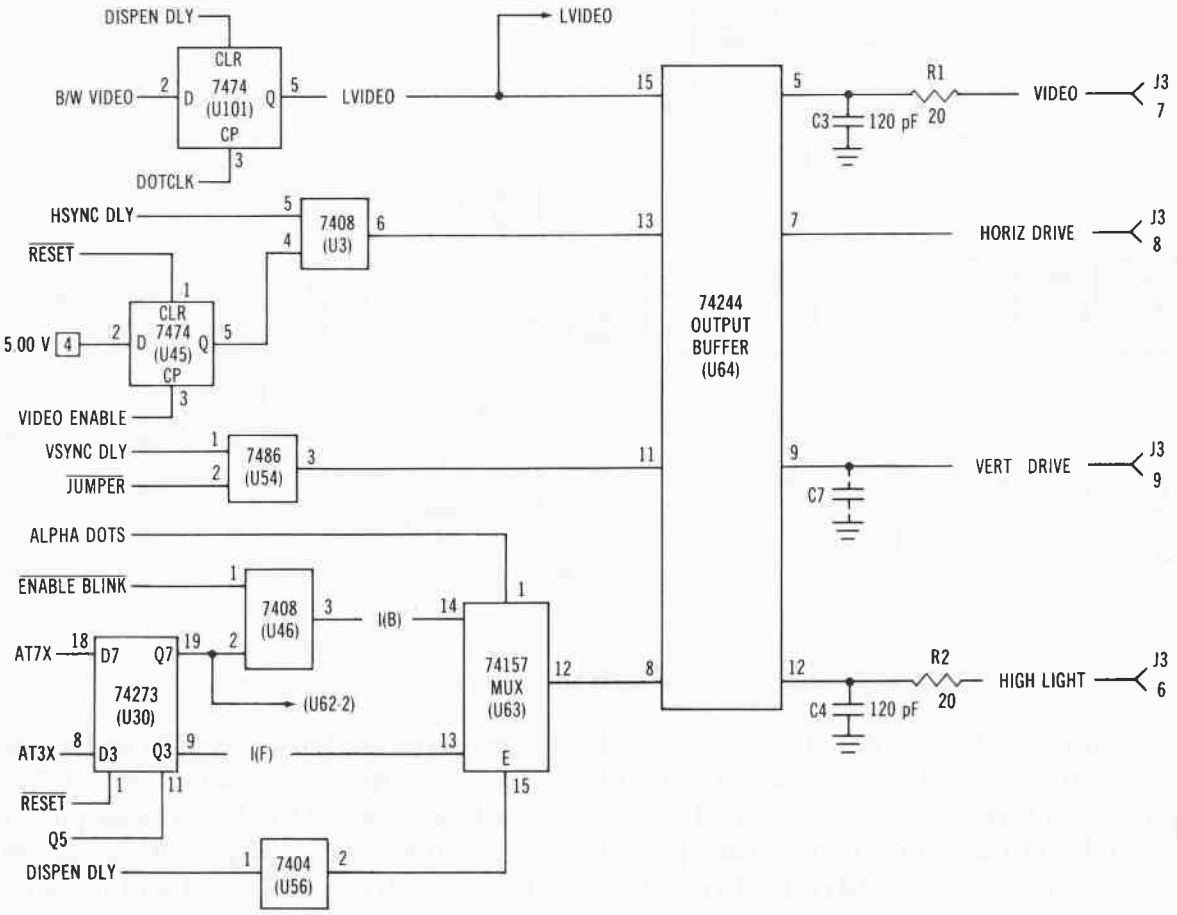


Fig. 2-81. Monitor adapter video output circuitry.

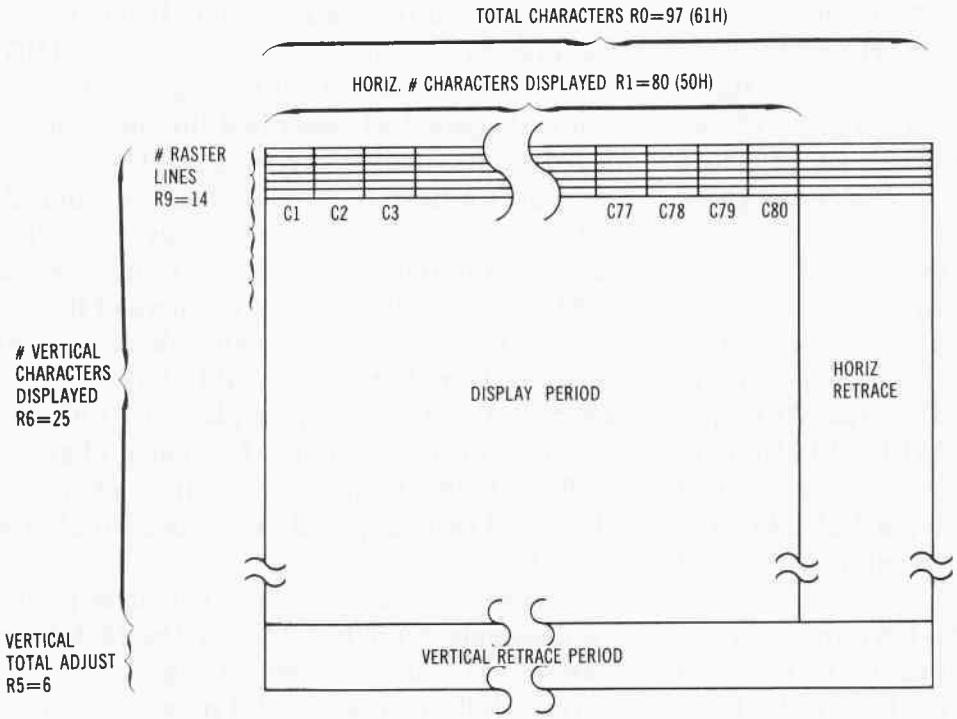


Fig. 2-82. 6845 CRTC register control of display screen.

over connector J3 are arranged such that each raster line is comprised of 97 nine-dot sequences. Each 9-dot packet represents one raster of the 14 needed to define a character box. As shown in the figure, two raster lines of no dots appear above the character, and three raster lines of no dots appear below the character. The additional raster lines below the A character provide room for an underline symbol (and for other characters, a descender).

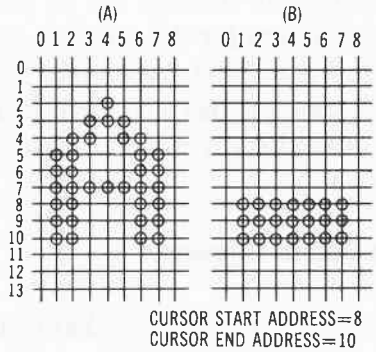


Fig. 2-83. Letter and cursor pixel activation within 7 by 9 of a 9 by 14 dot matrix.

Figure 2-83B shows one possible size of cursor display. The INT 10 interrupt command can be used to change the cursor size and position within the array box.

While the first raster line of dot information is shifted into the display unit, the horizontal and vertical drive signals control when the serial dot stream for a particular raster is complete and when the next raster line should begin. After 14 raster lines are shifted into the display unit, one row of characters is produced on the screen. The circuitry then begins to produce the first raster line in the second row of characters. And so the process goes until the last raster line has been sent out over connector J3 completing the last row of characters in the display.

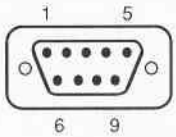
Color Graphics Adapter Card

The IBM color/graphics card can generate alphanumeric characters like the monochrome card. (Refer to COMPUTERFACTS CSCS2-B for this discussion.) It can also produce bit-mapped graphics. The board is designed to function with the IBM-produced color display unit. It can also drive other monitors and standard home television receivers.

Three video connections are on the board. One connection is an RCA jack for composite video output, as shown in Fig. 2-84. A second video interface is a 9-pin DIN connector for direct video drive to an RGB monitor. This output consists of separate red, green, and blue signals plus intensity and ground. A third output available on a 4-pin Berg strip provides a connection point for composite video, power, and ground. This interface is used to connect an RF modulator so you can use a standard television receiver with the PC.

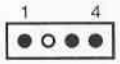
Pin	Signal
1	1.5-volt composite video
2	Chassis ground

RCA jack-composite video



Pin	Signal
1	Ground
2	Ground
3	Red
4	Green
5	Blue
6	Intensity
7	
8	Horizontal sync
9	Vertical sync

9-Pin din connector-color direct



Pin	Signal
1	+12 volts
2	-
3	Composite video
4	Logic ground

4-Pin Berg strip-RF modulator connections

Fig. 2-84. The video connections on the color/graphics adapter card.

A light pen connection is also available, but the mouse is the most popular input device complementing the keyboard.

To the system board, the color graphics adapter card looks like 9 I/O addresses plus 16 Kbytes of memory. The nine I/O addresses are given in Table 2-28.

Color Adapter Board 6845 CRTC

The color/graphics card contains the same type 6845 CRT controller as is used on the mono-

Table 2-28. Registers and Latches Directly Addressable on the Color/Graphics Adapter Card

Hex Address	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Register Function
3D0	1	1	1	1	0	1	0	Z	Z	0	6845 CRTC registers
3D1	1	1	1	1	0	1	0	Z	Z	1	6845 CRTC registers
3D4	1	1	1	1	0	1	0	Z	Z	0	6845 address register
3D5	1	1	1	1	0	1	0	Z	Z	1	6845 data register
3D8	1	1	1	1	0	1	1	0	0	0	Mode control register
3D9	1	1	1	1	0	1	1	0	0	1	Color select register
3DA	1	1	1	1	0	1	1	0	1	0	Status register
3DB	1	1	1	1	0	1	1	0	1	1	Clear lt pen latch
3DC	1	1	1	1	0	1	1	1	0	0	Preset lt pen latch

(Z = don't care condition)

chrome adapter card. On the color card, however, the CRTC must be reprogrammed each time graphics modes are changed. Figure 2-85 shows the pin assignments for the 6845 CRTC used on the color adapter card (see CF (CSCS2-B), page 2).

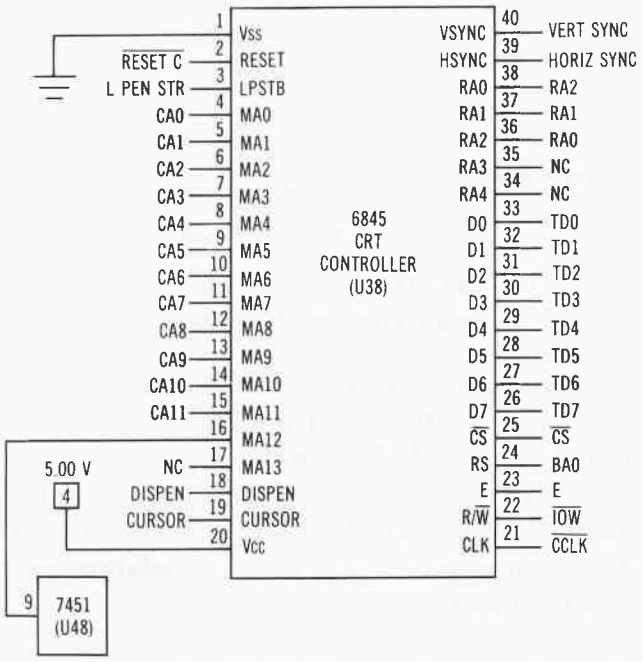


Fig. 2-85. Pin assignments for the 6845 CRT controller used on the color/graphics adapter card.

Reprogramming data enters and status data exits the CRTC as shown in Fig. 2-86. Hex addresses 3D0, 3D1, 3D4, and 3D5 access the 6845 I/O device. Pins A2 through A9 on the P2 expansion bus connector interface the 8-bit data bus (D0 through D7) of the 8088 CPU (U3) on the system board with the 74LS245 octal transceiver U66 on the color adapter card. Whenever DATA GATE is low and the output of 74LS08 2-Input AND (U41) is low, data flows from the transceiver data bus (TD0 through TD7) to the left out through the P2 connector as D0 through D7 data for the system board. When the pin 1 input to U66 goes high, data passes from the system data bus (D0 through D7) to the right out onto the transceiver data bus (TD0 through TD7).

6845 Chip Select Circuitry

Figure 2-87 shows the circuitry that generates the active low chip select signal CS* used by the 6845 CRTC U38. A binary 1 1 1 1 0 1 X X X X or X X X X X X 1 0 0 0 on the system address bus qualifies the inputs to 74LS32 OR gate U30 producing a low CS* signal out to pin 25 of U38. The first address qualifies 74LS08 AND U41, the 74LS138 decoder U18, and one input to 74LS32 OR U30 generating CS*. The latter address with bit A3 set qualifies the other OR input to U30.

6845 Enable Circuitry

The I/O read and write signals coming in from the system board expansion connector P2 are applied to a 74LS00 2-input NAND gate U15, as shown in Fig. 2-88. The ANDed output (pin 8) is applied to pin 1 of 74LS04 hex inverter U16 and the reset input (pin 13) and D input (pin 12) of 74LS74 dual D flip-flop U11. U11 is clocked by the system board signal CLK. The pin 9 Q output from U11 is the enable signal E that is applied to pin 23 of 6845 CRT controller U38.

6845 Clock Generation Circuitry

Figure 2-89 shows the circuitry that generates the control clock signal CCLK* that enters pin 21 of

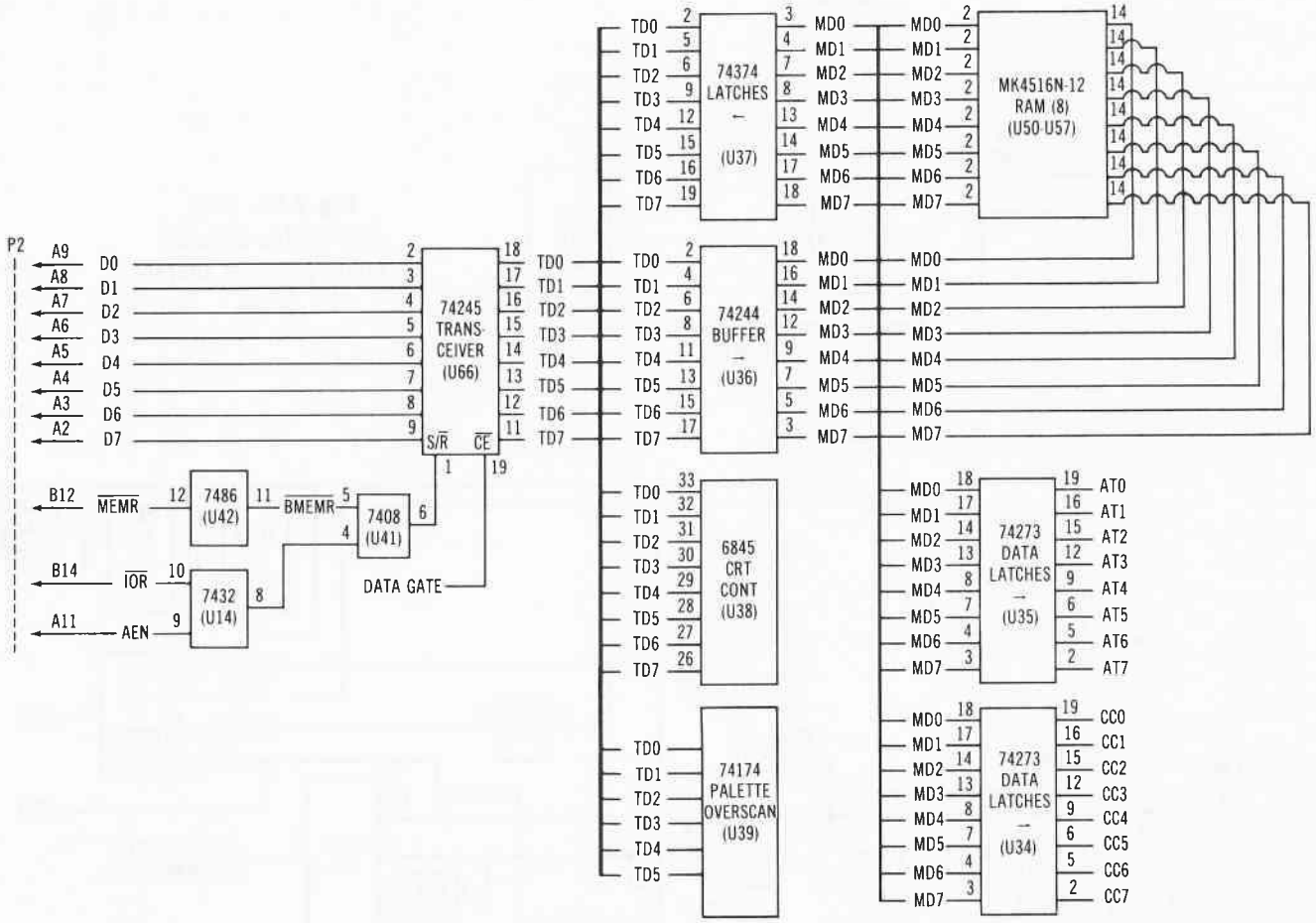


Fig. 2-86. Color adapter data bus circuitry.

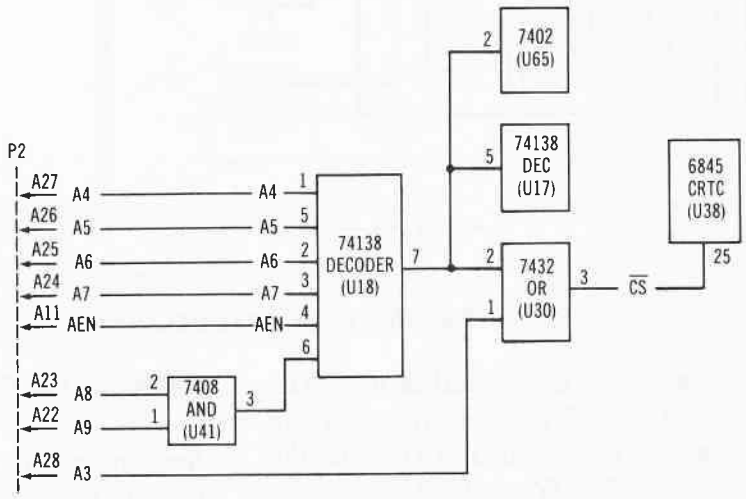


Fig. 2-87. Color/graphics adapter 6845 CRTC chip select circuitry.

6845 CRTC U38. A 14 MHz OSC signal from the expansion bus connector P2-B30 is inverted by 74LS04 U26 to produce the 14 MHZ clock signal. This signal is applied to the two 74LS174 timing generator chips U4 and U5 resulting in several special clock timing signals.

On the right side of Fig. 2-89 is a 74LS51 AND-OR-invert gate U27 that generates CCLK*

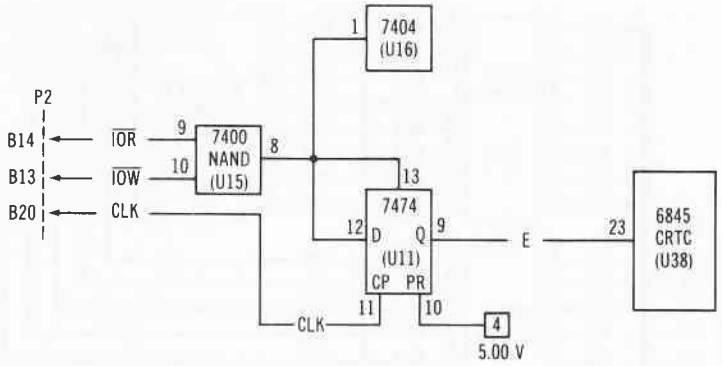


Fig. 2-88. 6845 chip enable circuitry on color/graphics adapter.

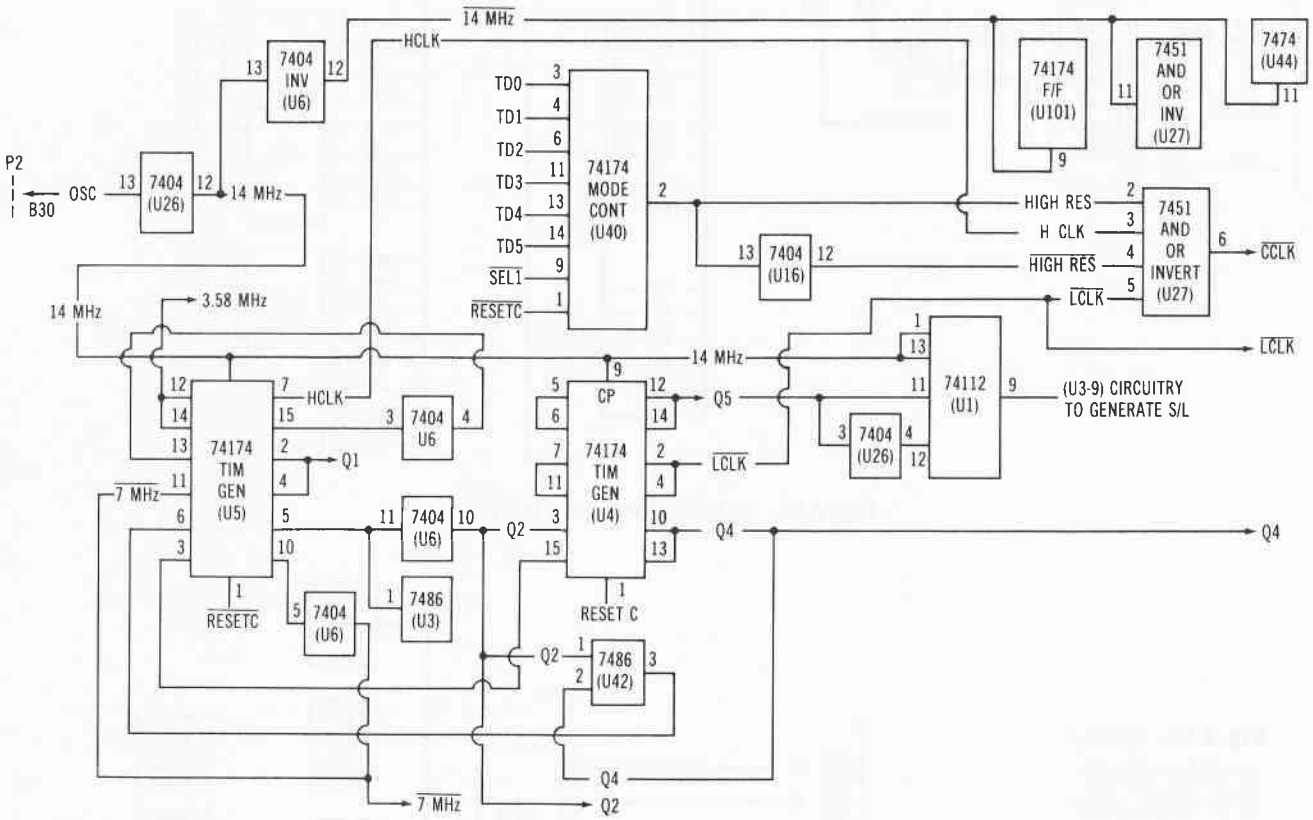


Fig. 2-89. Color 6845 CRTC clock circuitry.

whether high resolution is selected or not. The condition of HIGH RES is determined by the transceiver data bit TD0 on pin 3 of U40 and the pin 9 clock input SEL1*. When TD0 is high, the low-to-high transition of CCLK* transfers the TD0 high to its pin 2 HIGH RES output. HIGH RES is ANDed with the changing horizontal clock (H CLK) signal in U27 to produce a pulsing CCLK* for the 6845 CRTC U38.

Selecting the Color/Graphics Mode

Sending a byte of information from the system board to I/O address 3D8H causes the transceiver data byte to enter mode control register 74LS174 hex D flip-flop U40, as shown in Fig. 2-89. The output of U40 is high or low signal HIGH RES. When HIGH RES is logic high, it is ANDed with H CLK to produce an output CCLK*

that clocks at the H CLK rate. When HIGH RES is LOW, input pins 4 and 5 to U27 qualify another AND gate inside U27 to produce a CCLK* signal that clocks at the LCLK* rate.

This means that two primary modes of operation are possible: one a high resolution (all points addressable, or bit-mapped graphics) mode and the other a low resolution (alphanumeric) mode like the monochrome card. Within these two modes, additional submodes are available. Both 40-column by 25-line and 80-column by 25-line alphanumeric modes are available with characters defined in either a 7 by 7 dot font pattern or a 5 by 7 dot font pattern within an 8 by 8 dot matrix array. In both text modes, up to 16 foreground colors and 8 background colors can be used. With either character font size, a single line is available for descenders. Because of this, the characters are not as well defined as those produced by the monochrome card. In addition, underlining is not used.

Blinking, highlighting, and reverse video are available in the black-and-white mode only. Selective character blinking can also be controlled.

In the graphics mode, each pixel is independently addressable so many character shapes can be generated. Three bit-mapped graphics modes are available: a nonsoftware-operating-system-supported low resolution color graphics mode with 160 dot rows and 100 pixel columns (pixels controlled in groups of two dots high by two dots wide), a software-supported medium resolution color graphics mode with 320 pixel rows and 200 pixel columns (each pixel individually controlled), and a software-supported high resolution black-and-white only graphics with 640 pixels in a row and 200 pixels in a column. In high resolution graphics each pixel is individually controlled.

Color Selection

Medium resolution can make each dot one of four color configurations–one of 16 preselected background colors, and three other preselected colors, as shown in Chart 2-1. Two sets of three-color palettes are available: color set 1 with green, red, and brown, and color set 2 with cyan, magenta, and white. The background colors include the eight basic colors produced by combinations of red, green, and blue, and an additional eight lighter versions of the same colors possible by using the intensity control bit.

Chart 2-1. Possible Colors in Medium Resolution Graphics

Black	Gray
Blue	Light blue
Green	Light green
Cyan	Light cyan
Red	Light red
Magenta	Light magenta
Brown	Yellow
White	Bright white

COLOR SET C0	COLOR SET C1
Green	Cyan
Red	Magenta
Brown	White

The color set is selected by sending a byte to I/O address 3D9H. This address accesses color-select circuitry. When a byte of data is sent to address 3D9H, the first six bits determine the background and foreground colors in the alphanumeric, and medium and high resolution modes. Bit 5 selects the active color set in medium resolution graphics.

Colors Available in Alphanumeric Mode

The foreground (character) and background colors for both the monochrome and color/graphics adapters are defined by the attribute byte as described in Table 2-29. The monochrome adapter will only produce the colors shown in Chart 2-1. Any other code will result in unrecognizable white characters on a white background.

With a color graphics card installed, the display foreground and background colors can be determined by Table 2-30. Note that the combinations of red, green, and blue alone are used to define the background color for characters in color mode. Sixteen colors are

Table 2-29. Alphanumeric Code Defining Foreground and Background Color on Both Monochrome and Color/Graphics Adapter Cards

Attribute Byte		Background Color	Character Color
7 6 5 4	3 2 1 0		
BL R G B	I R G B		
Bkground	Foreground		
B 0 0 0	I 0 0 0	Black	Black
B 0 0 0	I 1 1 1	Black	White
B 1 1 1	I 0 0 0	White	Black
B 1 1 1	I 1 1 1	White	White

Table 2-30. Background and Foreground Colors Available on the Color Adapter Card

Background Colors		Foreground Colors	
RGB	*Color*	*RGBI*	*Color*
0 0 0	Black	0 0 0 0	Black
0 0 1	Blue	0 0 1 0	Blue
0 1 0	Green	0 1 0 0	Green
0 1 1	Cyan	0 1 1 0	Cyan
1 0 0	Red	1 0 0 0	Red
1 0 1	Magenta	1 0 1 0	Magenta
1 1 0	Brown	1 1 0 0	Brown
1 1 1	White	1 1 1 0	White
		0 0 0 1	Gray
		0 0 1 1	Light blue
		0 1 0 1	Light green
		0 1 1 1	Light cyan
		1 0 0 1	Light red
		1 0 1 1	Light magenta
		1 1 0 1	Yellow
		1 1 1 1	High intensity white

available for each character with blinking available on a per-character basis.

Monitoring the Status of the Video

Executing an 8088 I/O IN instruction from hex address 3DA causes the contents of 74LS244 status register U24 to be read out of the adapter board onto the system board through expansion slot connector P2, as shown in Fig. 2-90.

The circuitry works as follows. An I/O access is initiated by the 8088 CPU pulling IOR* low on pin 9 of 74LS00 NAND U15. Its high pin 8 output is inverted by 74LS04 U16 to place a low on pin 4 of 74LS138 decoder U17. Pin 4 is an active low enable input for U17. Its other active low input (pin 5) must also be low to produce an output from U17. Pin 5 gets its signal from output pin 7 from another 74LS138 decoder (U18). The address from the system board to the I/O circuitry is 3DAH. In binary, this is 1 1 1 1 0 1 1 0 1 0. Thus A9=1, A8=1, A7=1, A6=1, A5=0, A4=1, A3=1, A2=0, A1=1, and buffered address bit BA0=0. With these conditions, 74LS08 AND U41 is qualified producing a logic high out pin 3. Active low A5 and AEN inputs to 74LS138 decoder U18 enable this chip. The input combination of A4, A6, A7, and the U41 output on pin 6 of U18 cause the output pin 7 to go active low enabling the downstream decoder U17.

The input conditions BA0, A1, A2, and A3 on pins 1, 2, 3, and 6 of U17 cause output pin 13 to go low producing (STATUS SEL)*. This signal is passed to the pin 19 output enable of 74LS244 status register U24 allowing the condition of light pen Switch (L PEN SW)*, light pen strobe (L PEN STR), display enable (DISPEN*), and vertical sync delay (VERT SYNC DLY) on pins 13, 15, 17, and 11 to pass through U24 and output as transceiver data TD0 through TD3.

Figure 2-90 shows that TD0 through TD7 are passed through 74LS245 transceiver U66 and exit as data bus information D0 through D7. The program running in the 8088 CPU receives the byte of data and uses D0 through D3 to determine the status of the video circuitry. An active high bit 0 indicates that an I/O video memory access can be made without interfering with the video going to the display. Bit 1 high indicates that a positive transition from the light pen has set the light pen's trigger. This trigger is reset with an I/O OUT command to 3DBH or system power-on. Bit 2 reflects the status of the light pen switch. This bit is low when the switch is on. A high in bit 3 position indicates that the raster is in vertical retrace and screen-buffer updating can be initiated.

Memory Utilization

Two types of memory are mounted on the adapter card. An 8K character generator ROM

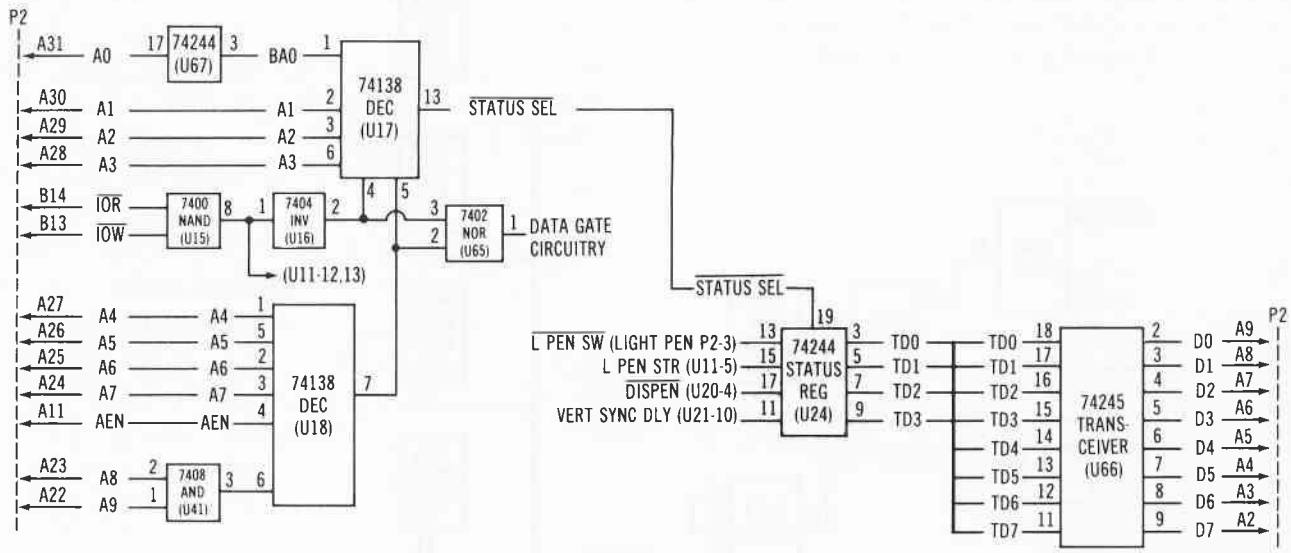


Fig. 2-90. Color status register circuitry.

Table 2-31. RAM Utilization for the Color/Graphics Adapter Card Modes

Mode	Amount of Memory per Screen	Screens of Storage
A/N (40x25)	1000 bytes character 1000 bytes attribute	8 total
A/N (80x25)	2000 bytes character 2000 bytes attribute	4 total
Low-resolution	(not supported in ROM)	
Medium-resolution	200 rows of 320 pixels, 4 pixels defined per byte = 16,000 bytes (C1-C0 per pixel)	1 (memory-mapped)
High-resolution	200 rows of 640 pixels, 8 pixels defined per byte = 16,000 bytes (Pixel P0-P7 per byte)	1 (memory-mapped)

contains dot patterns for 256 different characters in three different font styles. Two of the three font styles are used on the color card: a 7-high by 7-wide double-dot font, and a 5-wide by 7-high single-dot font. The desired font is jumper selected at connection P3. Inserting the jumper selects the single dot 5 by 7 font. Removing the jumper selects the double dot 7 by 7 font.

The other memory is a 16K byte dynamic RAM used to store character and color/attribute information. Table 2-31 lists the memory utilization for each of the video modes.

Alphanumeric data and bit-mapped graphics information are stored in 16K bytes of dynamic RAM. The I/O addresses reserved for this memory are from B8000H to B8FFFH. RAM ICs U50 through U57 are used to store the information which appears on the monitor screen. Storage and access function much like that on the monochrome adapter card. The memory on the color card is slower than the static RAM on the monochrome card so some blinking of the screen display can occur when performing a screen scroll function using the color card.

Color Adapter Board Reset Circuitry

Figure 2-91 shows the circuitry associated with the reset function on the color adapter board. RESET DRV from pin B2 of the system board expansion connector P2 is one input (pin 13) of 74LS86 quad 2-input XOR gate U68. The other input to U68 comes from the exclusive ORing of

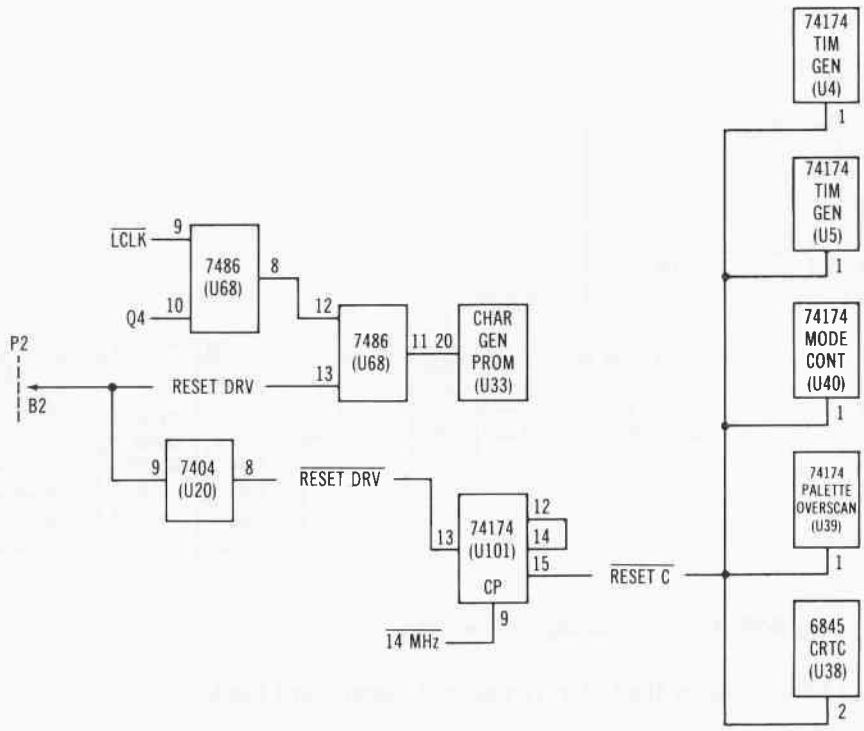


Fig. 2-91. Color reset circuitry.

LCLK* and Q4 in another gate of U68. The output of U68 (pin 11) is applied to pin 20 of character generator PROM U33.

The system board reset signal (RESET DRV) is also applied to 74LS04 hex inverter U20 to generate (RESET DRV)* on output pin 8. This signal becomes the D4 input to 74LS174 hex D flip-flop U101. The Q4 output on pin 12 is routed back into the D5 input (pin 14) to produce (RESET C)* on the Q5 output (pin 15). U101 is clocked by the 14MHz* signal that is generated elsewhere on the adapter board.

(RESET C)* is the primary reset signal for the board. It is connected to the pin 1 master reset input to four other 74LS174 hex D flip-flop chips, timing generators U5 and U14, the mode controller U40, and the palette overscan U39. This reset signal is also applied to the pin 2 reset input to 6845 CRT controller U38.

Video Output Signals

As described in Fig. 2-84, three video outputs are available on the color/graphics adapter: direct drive video and two composite video outputs.

Figure 2-92 covers the circuitry that develops the RGB and intensity direct drive video signals that are available at the edge connector P2. Color data is strobed through two 74LS153 color encoder multiplexers U9 and U10 to become red, green, blue, and intensity information onto output pins 7 and 9.

Multiplexer signals MUX A and MUX B determine the sequential outputs on pins 7 and 9. When the active enable inputs (STR) on pins 1 and 15 are low, the pins 7 and 9 outputs are determined by the select inputs MUX A and MUX B on pins 2 and 14. Table 2-32 lists the output for the various MUX A and MUX B input codes.

As shown in Fig. 2-93, the determination of the logic value for MUX A and MUX B can be quite complex. The alpha serializer data from 74LS166 U32 is ANDed with the pin 3 output from 74LS32 OR gate U14. Its inputs are the NANDed signals AT7, ENABLE BLINK, and CURSOR DLY and the pin 11 output from 74LS393 binary ripple counter U12. Counter U12 is clocked by VERT SYNC DLY. Pin 11 of U13 connects with input pin 4 of 74LS32 NOR U14. Its other input comes from the output of 74LS02 NOR gate U49. The inputs to U49 are the Q3 output of counter U12 and CURSOR DLY. NOR gate U14 combines its input with GRPH* and eventually

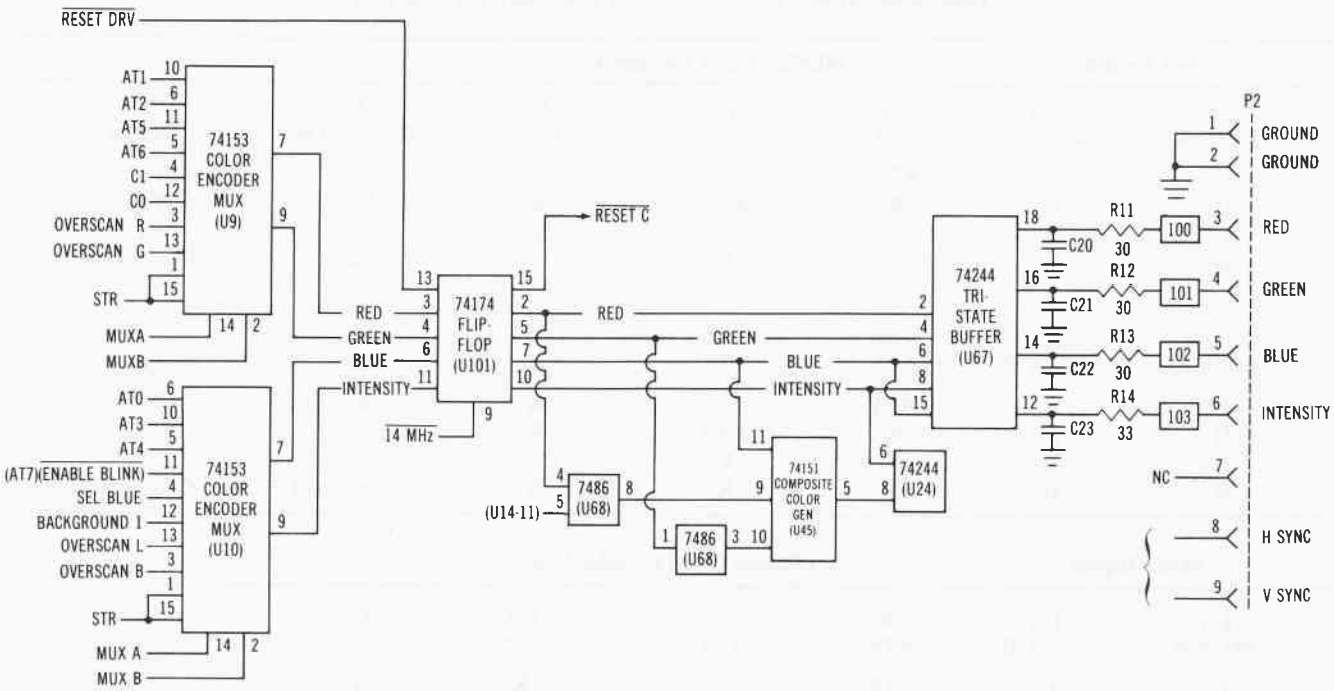


Fig. 2-92. Direct drive color circuitry.

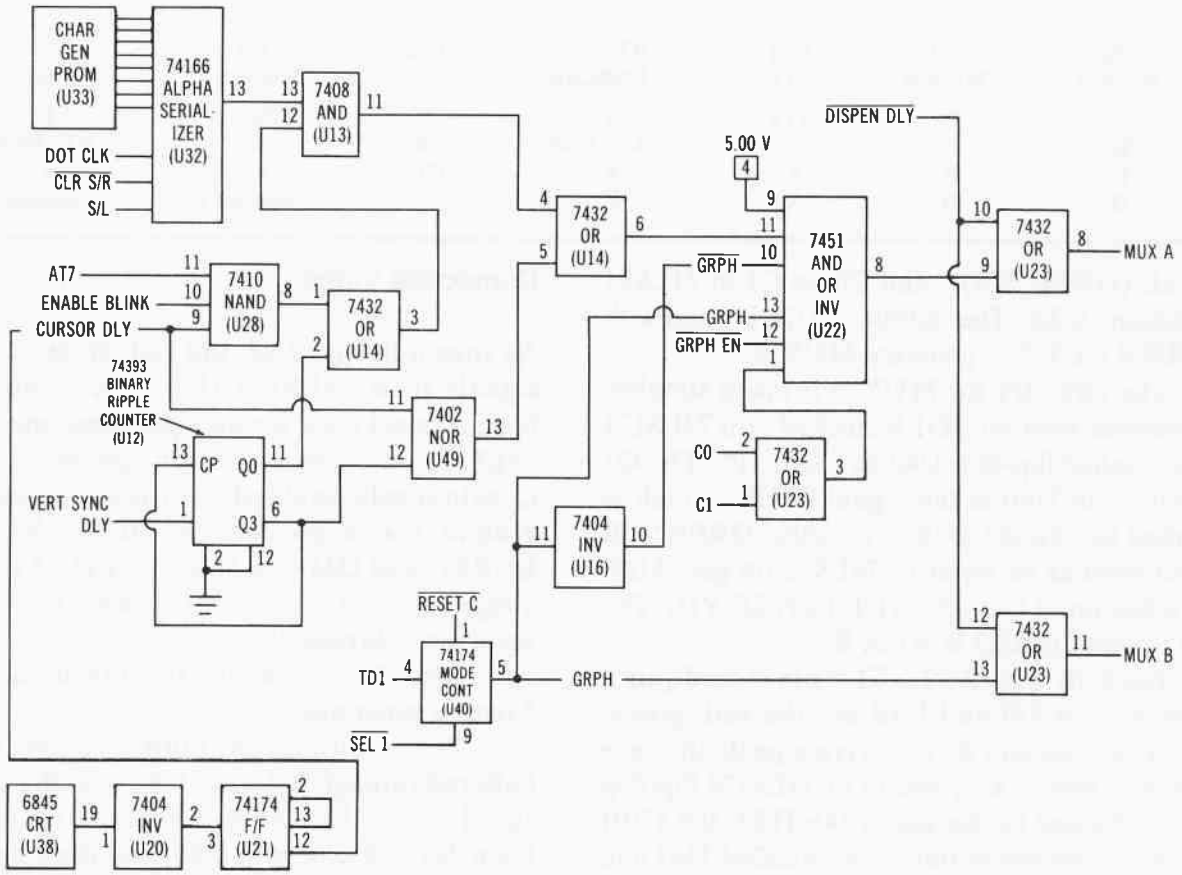


Fig. 2-93. Origination of MUX A and MUX B signals.

Table 2-32. Truth Table for Color Encoders U9 and U10

Select Inputs		74LS153 Color Encoder U9				
(14) MUX A	(2) MUX B	(6) AT2	(5) AT6	(4) C1	(3) Overscan R	(7) Output
L	L	AT2	X	X	X	AT2
H	L	X	AT6	X	X	AT6
L	H	X	X	C1	X	C1
H	H	X	X	X	Overscan R	Overscan R
(14) MUX A	(2) MUX B	(10) AT1	(11) AT5	(12) C0	(13) Overscan G	(9) Output
L	L	AT1	X	X	X	AT1
H	L	X	AT5	X	X	AT5
L	H	X	X	C0	X	C0
H	H	X	X	X	Overscan G	Overscan G
Select Inputs		**74LS153 Color Encoder U10**				
(14) MUX A	(2) MUX B	(6) AT0	(5) AT4	(4) Set Blue	(3) Overscan B	(7) Output
L	L	AT0	X	X	X	AT0
H	L	X	AT4	X	X	AT4
L	H	X	X	Set Blue	X	SET BLUE
H	H	X	X	X	Overscan B	Overscan B
(14) MUX A	(2) MUX B	(10) AT3	(11) AT7 ENBLNK*	(12) C1	(13) Overscan L	(9) Output
L	L	AT3	X	X	X	AT3
H	L	X	AT7.ENB	X	X	AT7.ENB
L	H	X	X	C1	X	C1
H	H	X	X	X	Overscan L	Overscan L

GRPH, (GRPH EN)*, and C0 OR C1 in 74LS51 AND-OR-Inv U22. The output of U22 is ORed with (DISPEN DLY)* to generate MUX A.

The circuitry for MUX B is much simpler. Transceiver data bit TD1 is clocked into 74LS174 mode control flip-flop U40 by (SEL 1)*. The Q1 output from U40 is the signal GRPH which is inverted by 74LS04 U16 to produce GRPH*. It is also used as an input to 74LS32 OR gate U23. The other input to U23 is (DISPEN DLY)*. The pin 11 output to U23 is MUX B.

Back to Fig. 2-92. The pin 7 and pin 9 outputs from U9 and U10 are the red, green, blue, and intensity direct drive signals that are applied to pins 3, 4, 6, and 11 of 74LS174 flip-flop U101. Clocked by the signal 14MHZ*, the U101 output signals are buffered by 74LS244 U67 and passed across individual RC networks to pins 3, 4, 5, and 6 of direct drive video connector P2.

Composite Video

As shown in Fig. 2-94, the red, green, and blue signals from 74LS174 U101 are combined to form a select code for an eight-input multiplexer 74LS151 composite color generator U45. Continuously enabled with pin 7 strapped to ground, U45 sequences the BLUE, YELLOW BURST, and U44 pins 9, 8, 6 (RED), 5 (CYAN), a logic high on input pin 12, and a logic low on input pin 4 through the chip.

Table 2-33 shows the truth table for the pin 5 output generation.

The multiplexed output from U45 is buffered through 74LS244 U24 with the intensity signal from 74LS174 flip-flop U101, delay signals from 74LS02 NOR gate U65, and the circuitry in the lower half of Fig. 2-94. The serial inputs to 74LS164 serial-in, parallel-out shift register U64

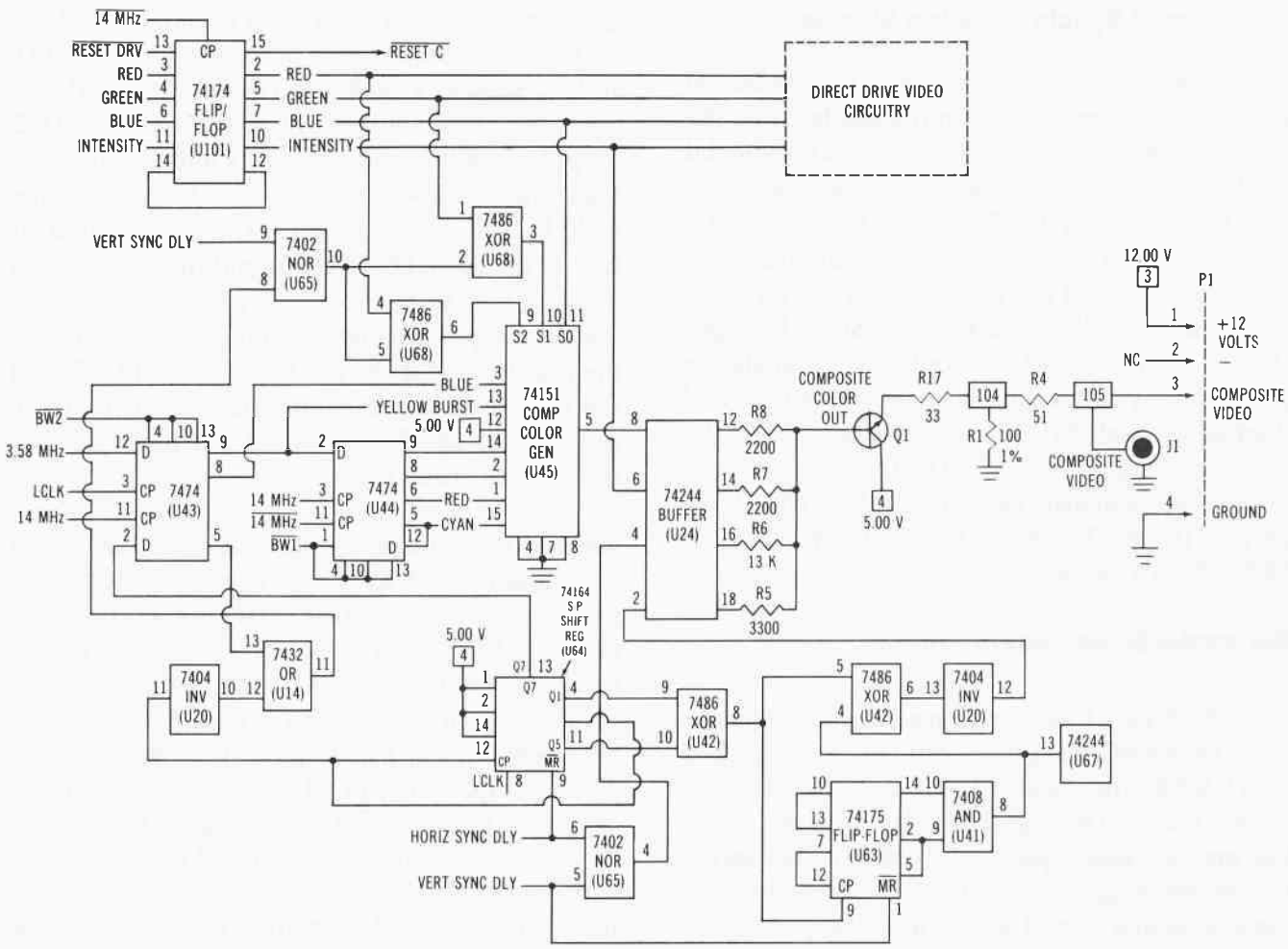


Fig. 2-94. Composite color circuitry.

are strapped high causing two of the eight outputs to become periodically high as clocked by the pin 8 input L CLK. Output pin 4 cycles between two clocks high then two clocks low in a 0 0 1 1 0 0 1 1 ... sequence. Every 32 clock cycles pin 11 toggles.

Table 2-33. Composite Color Generator Truth Table

(9) S2	(10) S1	(11) S0	(4) I0	(3) I1	(2) I2	(1) I3	(15) I4	(14) I5	(13) I6	(12) I7	(5) Output
L	L	L	I0	X	X	X	X	X	X	X	Ground
L	L	H	X	I1	X	X	X	X	X	X	Blue
L	H	L	X	X	I2	X	X	X	X	X	U44-8
L	H	H	X	X	X	I3	X	X	X	X	Red
H	L	L	X	X	X	X	I4	X	X	X	Cyan
H	L	H	X	X	X	X	X	I5	X	X	U44-9
H	H	L	X	X	X	X	X	X	I6	X	Yel Brst
H	H	H	X	X	X	X	X	X	X	I7	High

These two clocked outputs are passed through the XOR gate 74LS86 U42. The output of U42 is connected to another input (pin 5) to 74LS86 U42 and to the clock input to 74LS175 flip-flop U63. An ANDed output from U63 becomes the other input (pin 4) to U42. Its output gets inverted by U20 and passed to the 74LS244 buffer U24 as a timing signal that is combined with the other three inputs to U24 in the resistor-transistor-resistor circuitry on the output of U24. All four sequence signals from U24 are combined in composite color out transistor Q1 to produce the composite video signal that is available at pin 3 of composite video connector P1 or at the RCA jack J1.

Color Board Synchronization Signals

To lock-step the video signals with the horizontal and vertical sweep circuits in the display unit, the color/graphics adapter board generates special synchronization signals that are available at the 9-pin DIN connector J2. The 15.75 kHz horizontal drive and 60 Hz vertical drive signals from the color adapter card are generated by the circuitry shown in Fig. 2-95 (see also CF (CSCS2-B) pages 16, 17). Vertical and horizontal sync signals are generated by the 6845 CRT controller U38 and clocked through 74LS174 D flip-flop U21 by the pin 9 clock signal H CLK. The Q outputs of U21 become vertical sync delay (VERT SYNC DLY) on pin 10 and horizontal sync delay (HORIZ SYNC DLY) on pin 7.

Horizontal Drive Generation

The 15.75 kHz horizontal sync signal begins as HORIZ SYNC out pin 39 of U38. It is clocked by H CLK through U21 to become HORIZ SYNC DLY. This signal is applied to the active low master reset input (pin 9) of 74LS164 serial-to-parallel shift register U64. Because both of the AND inputs (pins 1 and 2) are strapped high, a logic high is clocked through the eight D-latches in U64 by input pin 8 clock signal LCLK. Outputs are taken from four of the eight stages: Q1 on pin 4, Q5 on pin 11, Q6 on pin 12, and Q7 on pin 13. The outputs of interest are the Q1 and Q5 stages. Pins 4 and 11 connect to 74LS86 XOR U42. When either Q1 or Q5 are high, U42 output pin 8 is high. When Q1 and Q5 are low, U42 pin 8 is low. When both Q1 and Q5 are high, U42 pin 8 is also low. The pin 8 output of U42 is the HORIZ SYNC signal that is buffered through U67 to connector J2 pin 8. Therefore, the clocking of a sequence of high inputs through the register under the clock control of LCLK and the master clear control of HORIZ SYNC DLY produces a 15.75 kHz signal out pin 8 of U42 through U67 to J2 pin 8.

Vertical Drive Generation—The 6845 CRTC pin 40 output (VERT SYNC) is clocked through U21 to become VERT SYNC DLY on U21 output pin 10. VERT SYNC DLY is applied to the master reset input to 74LS175 D flip-flop U63 (pin 1). Each time VERT SYNC (U38 pin 40) goes low, all the flip-flops in U63 are cleared. The Q0* output (pin 14) goes high applying a logic 1 to pin 10 of 74LS08 AND gate U41. Pin 4 of U63, the D3 input, is strapped to +5.00 volts causing the flip-flop to enter a high state upon the next low-to-high transition of the pin 9 clock input. Output Q3 (pin 2) connects to input D2 (pin 5). It also connects to pin 9 of U41. A logic high on both of the U41 inputs qualifies the AND

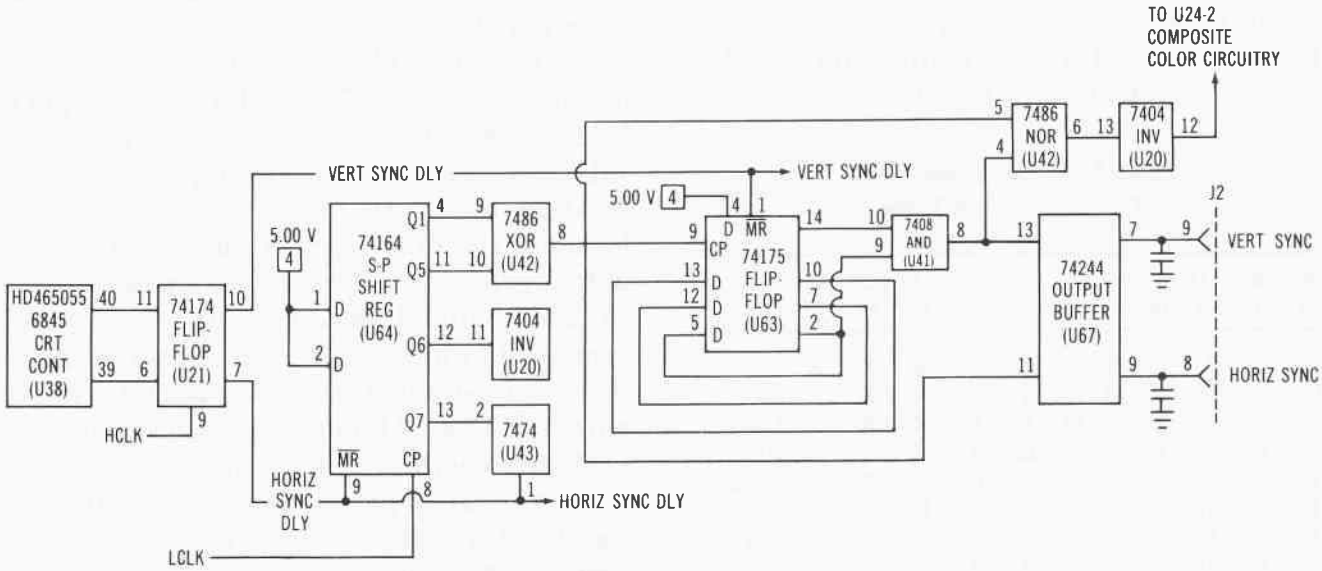


Fig. 2-95. Horizontal and vertical sync circuitry.

gate generating a high on pin 8. This high signal is buffered through 74LS244 U67 as a vertical sync signal at pin 9 of connector J2.

Meanwhile, back at U63, the pin 5 D2 input went high so on the next low-to-high clock transition, Q2 (pin 7) goes high. This output pin is connected to the D1 input pin 12. Output pin 10 (Q1) connects to D0 input (pin 13). And output pin 14 (Q3) connects to pin 10 of U41. Therefore, Q3 goes high and remains high for four clock cycles before Q3 returns low. Q3 remained high all during the cycle until another master reset pulse occurs on pin 1 momentarily clearing Q3 disabling pin 9 of U41. In this fashion, using the horizontal sync signal as a clock into pin 9 of U63, the occurrence of VERT SYNC DLY, and the four-step flip-flop of U63, the output of J2 pin 9 becomes a 60 Hz vertical sweep circuit synchronization signal.

FLOPPY DISK DRIVE INTERFACE

The IBM 5¼ inch diskette drive adapter that plugs into one of the expansion slots has connections for two internal and two external double density disk drives. The adapter works in the IBM System 34 double density format "Modified Frequency Modulation" (MFM) with write precompensation and an analog phase-lock loop for clock and data recovery. The adapter is buffered on the system I/O bus and uses DMA for data transfer to and from the storage disks in the drives. To the system board the diskette adapter (and hence the disk drives) looks like three I/O locations: an output register at location 3F2H, an input status register at location 3F4H, and a bidirectional data register at location 3F5H. (Refer to COMPUTERFACTS CSCS2-C for this discussion.)

The heart of the disk drive electronics is an NEC microprocessor D765 or equivalent floppy disk controller (FDC) labelled U6 in Fig. 2-96. The FDC controls the operation of up to four double density disk drives. It simplifies the architecture of write precompensation and phase-locked loop read circuitry. As the figure shows, U6 is comprised of a data bus interface buffer, DMA read/write control logic, a main status register and data register, serial interface control for the drives, and drive interface control for reading, writing, and head control.

Handshaking signals enable DMA interface with the DMA controller on the system board. The FDC will operate in both DMA and non-DMA modes. In the non-DMA mode, U6 generates an interrupt to the 8088 CPU on the system board every time a data byte is to be transferred. For DMA operation, the 8088 CPU loads a command into the FDC and data transfer occurs under control of the FDC and the DMA controller.

FDC U6 can execute up to 15 high level disk commands including formatting a disk, seeking, writing data, writing a deleteted track, reading data, reading ID, reading a track, reading deleted data, and sensing interrupt and drive status. Multiple 8-bit bytes specify which operation is to be performed.

Data synchronization and error checking is performed automatically by U6 for reliable data storage and retrieval. External circuitry is used to generate master clock and write clock signals for the FDC and for data separation during read operations. FDC U6 also generates control signals for start-up and data separator base frequency selection. Because the 765 FDC can interface to a large number of disk drives, the BIOS software sends commands to U6 to specify the track stepping rate, and the head load and unload times.

Figure 2-97 describes the pin assignments for the FDC designed into the IBM PC diskette adapter board. Some pins are not used. Pin 26 (MFM) is a flag output identifying the mode configuration of U6. Because the board operates in MFM all the time, this status condition is not required; therefore, pin 26 is not used.

Pins 28 and 29 (unit select 0, 1) are available to select one of four floppy disk drives in a system. A different drive select design is used on the diskette adapter board.

Finally, pin 36 (HDL) is not used to cause the drive read/write head to contact the disk. The drives used with this adapter have their own

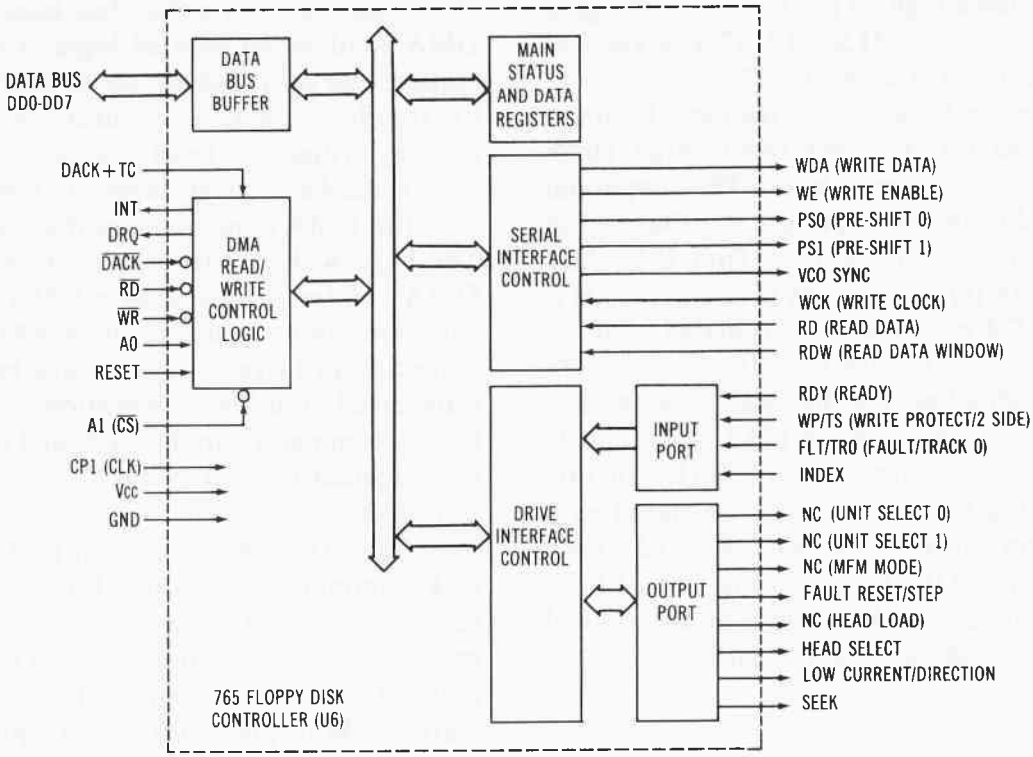


Fig. 2-96. The 765 FDC internal block diagram.

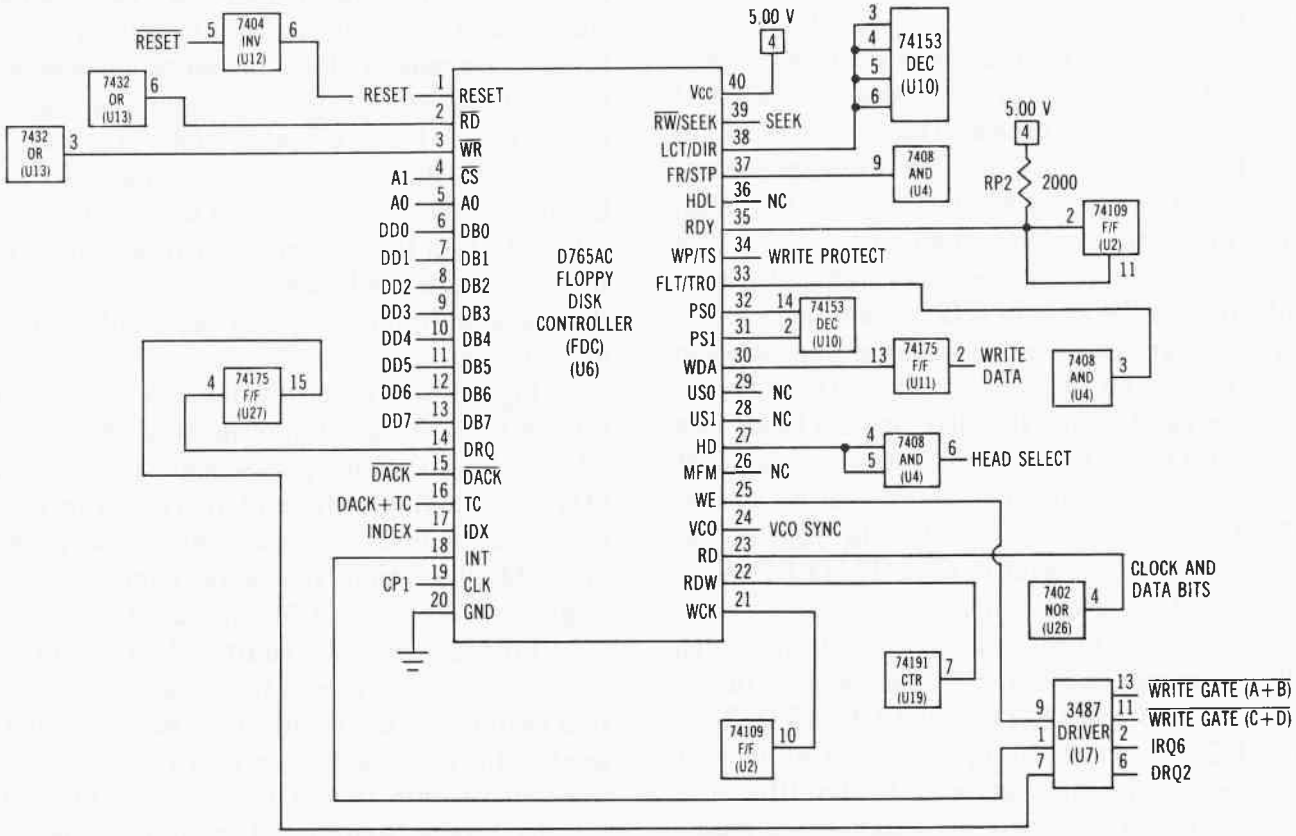


Fig. 2-97. Floppy disk controller pin assignments.

head-positioning system that moves the magnetic head so it contacts the desired track on the disk. A stepper motor and band assembly in the drive uses one-step rotation to cause a single-track linear movement of the head. Because no other system or operator intervention is needed during normal operation, a signal out pin 36 is not required.

Table 2-34 describes the function of each 765 FDC pin.

FDC Operation

The main status and data registers can be accessed by 8088 CPU using addresses 3F4H and 3F5H. The 8-bit read-only main status register contains information related to the condition of the FDC. This register can be accessed at any time. The 8-bit data register is actually a window into one of four registers in a stack. The stack stores data, commands, parameters, and disk drive status information. Bytes of data are read out or written into the data register to program the drive electronics or to pull out status information after execution of a command. Table 2-35 describes how the registers are accessed.

Main Status Register

Data is read from or written to the FDC registers by the combination of A0, RD*, WR*, and A1 (chip select). The eight bits in the main status register byte are defined, as shown in Table 2-36. A1 must be active low to enable A0, RD*, and WR* control of the registers.

FDC Operational Phases

Multibyte transfers of information between the 8088 CPU and the 765 FDC during disk drive command initiation and post-execution. A command consists of a command phase, an execution phase, and a result phase. Data byte transfer must occur in a specified order. The command code is sent first, followed by the other bytes in a prescribed sequence. A data book describing the 765 FDC lists the byte sequence required.

During the command phase, U6 receives the information necessary to perform an operation. The executing program transfers disk operation information to the FDC data register. This occurs after a system reset and after completion of a previous command.

During the execution phase, the FDC performs the instructed operation. This phase ends when the last byte of data has been transferred (as signaled by TC from the 8237 DMAC to the FDC) or the occurrence of an error.

After the disk operation has been executed, the FDC enters the result phase. In this phase, status and housekeeping information is available to the 8088 CPU. All the bytes (typically seven) available in the result phase must be read to complete the read data command. A new command will not be recognized until all the seven bytes have been read out. After the CPU reads the FDC data register information, U6 reenters the command phase again and is ready to accept another command.

Writing data to or reading data from the FDC data register during the command and result phases is initiated with the 8088 accessing the FDC main status register to determine if the data register is available. Bits 6 and 7 must be low and high respectively before a command byte can be written into U6. During the result phase, bits 6 and 7 must both be high to enable reading from the FDC main status register.

Data transfers to, or from, the disk drive occur in the execution phase. During DMA transfers, a DRQ signal is generated out pin 14 of U6. This signal is passed through 74LS175 flip-flop U27 and out pin 6 of 3487 driver U7 as DRQ2. DRQ2 signals the 8088 that a DMA request has been made. The 8237 DMA controller on the system board responds by generating DMA acknowledge 2 (DACK2*). This signal becomes DACK* and DACK&TC. DACK* enters pin 15 of U6 signaling acknowledgment of the request. RD* or WR* is also generated by the 8237 DMAC. During the last transfer of data, the DRQ signal is reset. After the last data transfer, pin 18 INT becomes active high producing IRQ6 out pin 2 of U7.

Table 2-34. 765 Floppy Disk Controller Pin Functions

Signal	Pin	I/O	Function
RESET	1	I	Reset: Places FDC in idle state. Pulls output lines low. Doesn't clear last specify command.
RD*	2	I	Read: Active low signal that allows data transfer from FDC to I/O data bus. Disabled when CS* high.
WR*	3	I	Write: Active low signal allowing data transfer to FDC from data bus. Disabled when CS* high.
A1	4	I	Address Bit 1 (chip select): Active low address bit signal that allows RD* and WR* to be enabled.
A0	5	I	Address bit 0 (data/status select): Determines if data register (A0=1) or status register (A0=0) contents can be passed to data bus.
DD0-DD7	6-13	I/O	Data bus: Bidirectional data bus interface that is disabled when CS* is high.
DRQ	14	O	DMA request: Active high signal used to generate DRQ2 DMA request to 8088 CPU.
DACK*	15	I	DMA acknowledge: Active low when DMA cycle is active and DMAC on system board is controlling data transfer.
DACK&TC	16	I	DMA acknowledge and terminal count: The active high result of combining the DACK2* and TC signals from the system board to indicate the end of a DMA transfer. Ends read/write/scan commands in DMA or interrupt mode.
INDEX	17	I	Index: Active high to indicate the head has just detected the beginning of a new disk track.
INT	18	O	Interrupt: An FDC interrupt request that produces IRQ6 for the system board circuitry.
CPI	19	I	CPI (clock pulse): 4 MHz single phase square wave clock used to pulse the FDC circuits.
GND	20		Ground: DC power return.
WCK	21	I	Write clock: Sets the floppy disk drive write data rate to a 250 ns 1 MHz pulse. Enabled for both read and write operations.
RDW	22	I	Read data wimdow: Phase-locked loop generated signal used to sample data from disk drive.
RD	23	I	Read data: Composite data from drive that contains clock and data information.
VCO SYNC	24	O	VCO Sync: Inhibits voltage controlled oscillator in phase-locked loop when low. Enables VCO in PLL when high.
WE	25	O	Write enable: Used to generate (WRITE GATE A&B)* and (WRITE GATE C&D)* signals that enable write data into the disk drive.
MFM	26	O	Modified frequency modulation: Status output—not used.
HD	27	O	Head (select): Selects magnetic head 1 when high, head 0 when low.
US0, US1	28, 29	O	Unit select 0, 1: not used.
WDA	30	O	Write data: The serial clock and data that is output to disk drive electronics.
PS0, PS1	31, 32	O	Pre-shift 0, 1: Write precompensation status signals that determine early, late, and normal timing signals.
FLT/TR0	33	I	Fault/track 0: Senses drive fault condition in read/write mode and track 0 condition in seek mode.
WP/TS	34	I	Write protect/two side: Senses write protect status in read/write modes and two side media in seek mode.
RDY	35	I	Ready: Tied high through RP2 to indicate disk drive always ready to send or receive data.
HDL	36	O	Head load: not used.
FR/STP	37	O	Fault reset/ step: In read/write mode this signal resets fault flip-flop in the disk drive. FR pulse issued at beginning of each read/write command. In seek mode, FDC (U6) outputs step pulses to move the head to another cylinder track.
LCT/DIR	38	O	Low current/direction: In read/write mode this signal lowers write current on inner tracks; in seek mode it determines the direction the head will move when it receives a step pulse.
RW*/SEEK	39	O	Read write/seek: Specifies read/write mode when low and seek mode when high.
Vcc	40		Vcc: +5 volt DC power.

IRQ6 indicates that the execution phase has ended and the result phase is beginning. It causes a program jump to an interrupt handling routine that examines the result bytes and resets the interrupt request. The terminal count (TC) output from the 8237 DMAC enters the drive adapter board and is ANDed with an inverted DACK2* signal to produce a DACK&TC input to pin 16 of U6. This indicates that the byte of data initialized for DMA count is now being transferred.

During non-DMA, transfer request bit 7 in the main status register and the INT signal on 765 FDC pin 18 are both high. INT becomes IRQ6 for the interrupt-driven IBM PC system. Master request (bit 7) is used in software controlled architectures where polling is used to determine the status of the system. When IRQ6 is received by the 8088 CPU, it responds by reading data from U6 (RD* pulled active low) or writing data to U6 (WR* pulled active low). This disables the transfer request. When a transfer request has been received by the 8088 CPU for the last data byte, the system board causes a high to occur on pin 16 (DACK&TC) of U6. This occurs before the last data byte has actually been sent (or received).

Table 2-35. Decoding for Register Access

A0	RD*	WR*	Function
0	0	1	Read main status register
1	0	1	Read data register contents
1	1	0	Write data register

* (all other code combinations are illegal)

Once pin 16 of U6 goes high, the FDC stops requesting data transfers. U6 will continue to read from or write data to the disk until the end of the current disk sector is reached. Data read from the disk after TC occurs will be discarded. A cyclic redundancy check will be made. During a disk write, once terminal count has been reached, data output shifts to zeroes and the remainder of the sector is filled with zero bytes.

Table 2-36. Main Status Register Bit Definition

Bit	Name	Description
0	DRIVE 0 BUSY	Drive 0 is in seek mode.
1	DRIVE 1 BUSY	Drive 1 is in seek mode.
2	DRIVE 2 BUSY	Drive 2 is in seek mode.
3	DRIVE 3 BUSY	Drive 3 is in seek mode.
4	FDC BUSY	FDC read/write command in proces.
5	NON-DMA MODE	FDC in non-DMA mode. Set during execution phase. Low when phase has ended.
6	DATA I/O	Direction of data transfer for data register and FDC. high = transfer from data register to 8088 CPU. Low = transfer from 8088 CPU to data register.
7	MASTER RQST	Indicates data register ready to send or receive data. Used with bit 6 to perform handshaking of "ready" and "direction" with 8088 CPU.

FDC (U6) Electronics

FDC U6 performs six functions on the disk drive adapter board: One, it manages the selection of up to four disk drives.

Two, it controls track selection by issuing timed step pulses that move the magnetic head from its current location over a track cylinder to another specified cylinder for data reading or writing. FDC U6 stores each cylinder number being accessed by the head and computes the stepping distance to the next cylinder. It also manages the head select signal to activate the correct side of the diskette.

Three, it monitors the track data until it senses a requested sector (sector selection). Seek time is specified at 6 ms track to track.

Four, it controls head loading at 35 ms and causes the head to hold for 15 ms settling time before writing or reading, and head loading after read/write operations. Therefore, the head is loaded 35 milliseconds before a read/write access. Following access, the head is held 15 milliseconds before reading or writing. The motor is held powered up for 250 ms before head cylinder movement occurs.

Five, the FDC outputs the READ DATA composite signal (clock and data) into a data separation circuit at 500K bits/second for the double-density disks so the clock and data can be converted into two signal streams. The serial data is also assembled into 8-bit bytes and transferred to memory every 16 microseconds. Data separation is achieved using a phase-locked loop (PLL). After synchronizing with the data stream, the separation logic provides a data window to the FDC that differentiates data from clock information. FDC U6 uses this window to reconstruct the data that was previously recorded on the disk.

Finally, the FDC electronics includes cyclic redundancy check (CRC) circuitry to check for standard soft errors. As data is written onto the disk, a 16-bit CRC value is computed and also stored on the disk. When the data is later retrieved off the disk, a CRC value is computed on the read data and compared to the stored CRC value of the original data. If a match occurs, all is well. A mismatch results in a CRC error and a read retry by the system.

Disk Drive Electronics

Inside each disk drive connected to the adapter board are circuits that translate digital command signals into electromechanical actions (for example, drive selection, head movement, and head loading) and that sense and provide disk or drive status (for example, ready to read/write, write fault, and write protect) back to the adapter board circuitry. There is also analog circuitry that senses, amplifies, and shapes data pulses read from, or written to, the disk by the magnetic head.

Table 2-37 describes the adapter board input and output signals.

Table 2-37. Disk Drive Adapter Input and Output Signals

Signal	Description
Outputs:	
DRIVE SELECT (A & B) (P2-12,14)	Enables drive I/O for a selected drive. All other drivers and receivers (except motor enable) are disabled.
MOTOR ENABLE (A & B) (P2-10,16)	Controls the spindle motor in the selected drive. Pulling one of these lines active low starts the specified drive motor.
STEP (A & B) (P2-20)	Causes the selected drive read/write head to move one cylinder track in or out for each pulse depending on the direction selected by the signal on P2 pin 18.
DIRECTION (A & B) (P2-18)	Causes the head to move one cylinder toward the spindle for each step pulse if active high and one cylinder away from the spindle if active low.
SEL HD 1 (A & B) (P2-32)	Selects the upper head (head 1) when active low. Selects head 0 when high.
WRITE DATA (A & B) (P2-22)	Causes a flux change to be recorded on disk for each low to high transition while write enable is high.
WRITE GATE (A & B) (P2-24)	Disables write current in head unless active low.
Inputs:	
INDEX (A & B) (P2-8)	A pulse from the selected drive each time the index mark is detected during disk revolution.
READ DATA (A & B) (P2-30)	Pulse occurs on this line for each flux change sensed by read head of selected drive.
TRACK 0 (A & B) (P2-26)	Active when the read/write head of the selected drive is over track 0.
WRITE PROTECT (A & B) (P2-28)	Pulled active by the selected drive if a write-protected disk is detected in the drive.

Data Recording Technique

MFM data recording is used by the double density disk drive circuitry for encoding and decoding data stored on floppy disks in the IBM PC system. In MFM encoding data bits are written inside a bit cell time. During the write data execution phase, data transfers between the 8088 CPU and the 765 FDC occur every 15 microseconds. The presence of a data bit represents a binary 1; the absence of a bit represents a binary 0. The data bits are written

in the center of a bit cell with a clock bit written at the leading edge of the bit cell only if no data bit was written in the previous bit cell and no data bit is to be written in the present bit cell. This leads to a somewhat complex looking data/clock pattern but a denser data storage capability.

Disk Format

Each of the soft-sectored disks used in a PC disk drive has 40 tracks (cylinders) divided into 8 sectors per track (9 with DOS 2.0 and 2.1 operating systems) and 512 bytes per sector. This yields 327,680 bytes per disk in 8 sectors per track systems and 368,640 bytes per disk in 9 sectors per track systems.

Each of the 40 tracks on a floppy disk are partitioned into five sections, as shown in Table 2-38.

Each sector in a track is comprised of four fields as described in Table 2-39.

Table 2-38. Partitioning of a Track

Section	Function
Pre-index gap	Gap 5—written during formatting.
Index address marks	A unique code indicating the beginning of a data track. One index mark is written on each track during formatting.
Post index gap	Gap 1—used during read/write operations to synchronize data separator logic with data to be read from ID field of first sector. Writ-ten only during formatting.
Sectors	Four fields of data written during formatting and repeated for each sector on the disk. One field holds the data to be stored during write operations.
Final gap	Gap 4—written during formatting. Extends from last physical data field on track to physical index mark.

Table 2-39. Sector Field Partitioning

Field	Function
Sector ID field	Consists of seven bytes written during formatting. Provides sector identification to FDC. Byte 1 is FDC-provided unique coding ID address mark specifying the beginning of ID field. Bytes 2, 3, and 4 are cylinder, head, and sector addresses provided by the disk operating system. Byte 5 is the DOS-supplied sector length code. Bytes 6 and 7 are 16-bit CRC character value for ID field. This value is computed by the FDC from the data in the first five bytes.
Post ID field gap	Gap 2—written during formatting. During later write operations, drive write circuitry is enabled during gap time. Trailing bytes of gap rewritten each time sector is written into. During read operations, trailing bytes of gap used to synchronize data separator logic with upcoming data field.
Data field	Length determined by operating system during formatting. Byte 1 is data address mark specifying beginning of data field. Last two bytes comprise CRC character.
Post data field gap	Gap 3—written during formatting. Separates preceding data field from next physical ID field on track. Contains programmable number of bytes. Following write operation, drive write logic is disabled during Gap 3. Size large enough to contain disk surface data discontinuity during drive write current turn on and off and to contain synchronization field for next sector's upcoming ID field.

System Board Interface

With an understanding of how the disk drive and disk adapter circuitry generally function, it's appropriate to take a closer look at how the IBM adapter board implements MFM recording and data retrieval. Figure 2-98 shows the system board expansion interface to the circuitry on the disk drive adapter board.

Ten address bits are connected to the adapter board to access the three I/O location registers–3F2H for the 74LS245 digital output register (U30) in the lower left of Fig. 2-98; 3F4H for the FDC main status register; and 3F5H for the FDC data register. For all three of

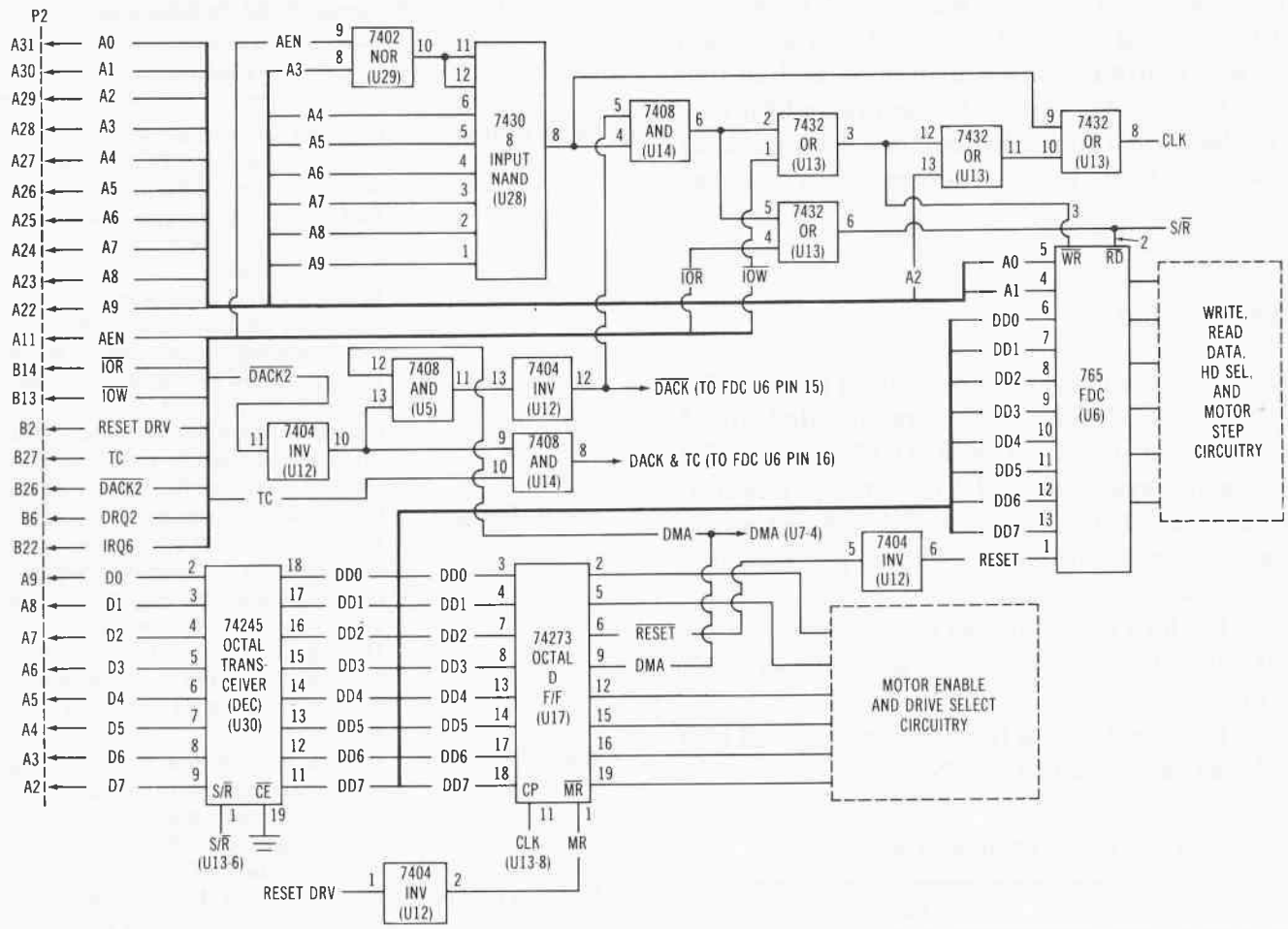


Fig. 2-98. System board to drive adapter board interface circuitry.

these addresses, address bits A4 through A9 are high and bit A3 is low.

Digital output register U30 is a bi-directional octal transceiver with direction of data flow controlled by the S/R* input on pin 1. This control signal is derived by the circuitry in the upper part of Fig. 2-98. Address bit A3 and address enable signal AEN are applied to the 74LS02 NOR gate U29 to produce the pin 11 and 12 inputs to 74LS30 8-Input NAND U28. The other six inputs to U28 are A4 through A9 directly off the address bus. The pin 8 output of U28 is ANDed with DACK* from U12 and applied to two separate inputs to 74LS32 U13. The top input (pin 2) is used to generate an input write control signal to pin 3 of 765 FDC U6. It is also used to generate a clock signal (CLK) that pulses data through 74LS273 octal D flip-flop U17 and into the motor enable and drive select circuitry. CLK also causes U17 to produce the DMA and RESET* pulses used elsewhere in the circuitry.

The other half of U13 receiving the output from U14 is on pin 5. This input combines with IOR* from the P2 expansion connector (pin B14) to produce S/R* out pin 6. This signal controls the direction of data flow through U30. It also acts as an active low control signal for the 765 FDC U6. This signal allows the transfer of data from U6 to the expansion I/O bus. It is disabled internally in U6 when A1, the FDC chip select (pin 4) is high.

The other two addresses 3F4H and 3F5H vary by the condition of bit A0, the DATA/STATUS SELECT input to U6 (pin 5). When A3 is low, and A4 through A9 are high (0 0 1 1 1 1 1 1 0 x x x), an active high A2 causes the generation of CLK out pin 8 of U13 (top right in

Fig. 2-98). CLK pulses the data bus to output DD3 (pin 8 of 74LS273 octal D flip-flop U17 as a pin 9 DMA signal. DMA is tied to control pin 4 of 3487 driver U7. This makes the qualifying code (0 0 1 1 1 1 1 1 0 1 x x). The condition of address bit A0 determines if the FDC data register (A0=high) or main status register (A0=low) will be passed to the data bus out pins 6 through 13 of U6.

Adapter Board Clock Circuitry

Figure 2-99 shows the circuitry on the disk drive adapter board that generates the CLK, CPI, 16 MHz, and 2 MHz clock signals. As described in the previous section, address bits A2 through A9 (CF (CSCS2-C), pages 4 and 21) are used to generate the CLK signal output from pin 8 of 74LS32 OR gate U13. The CLK signal is used to drive 74LS273 octal D flip-flop U17.

A 16 MHz voltage controlled oscillator is used to generate the remaining three clock signals (16 MHz, 2 MHz, and CPI) in the clock/data separator circuitry. Exiting pin 3 of U1, 16 MHz is passed across connector P3 to the pin 4 clock input of 74LS09 flip-flop U2 and pins 1 and 13 of 74LS112 dual JK flip-flop U22. The active high input of U2 is passed out the Q* output (pin 7) to clock a 74LS93 4-bit counter U3. U3 generates a 4 MHz CPI signal and a 2 MHz signal. CPI is used to clock 765 FDC U6 and a 74LS175 flip-flop U11 in the precompensation circuitry. The 2 MHz pulse stream out pin 9 of counter U3 clocks 74LS175 flip-flop U27 to pass the DMA request signal DRQ on input pin 4 out pin 15 as the DMA request 2 signal DRQ2 that is coupled through pin B22 of I/O connector P5 onto the system board.

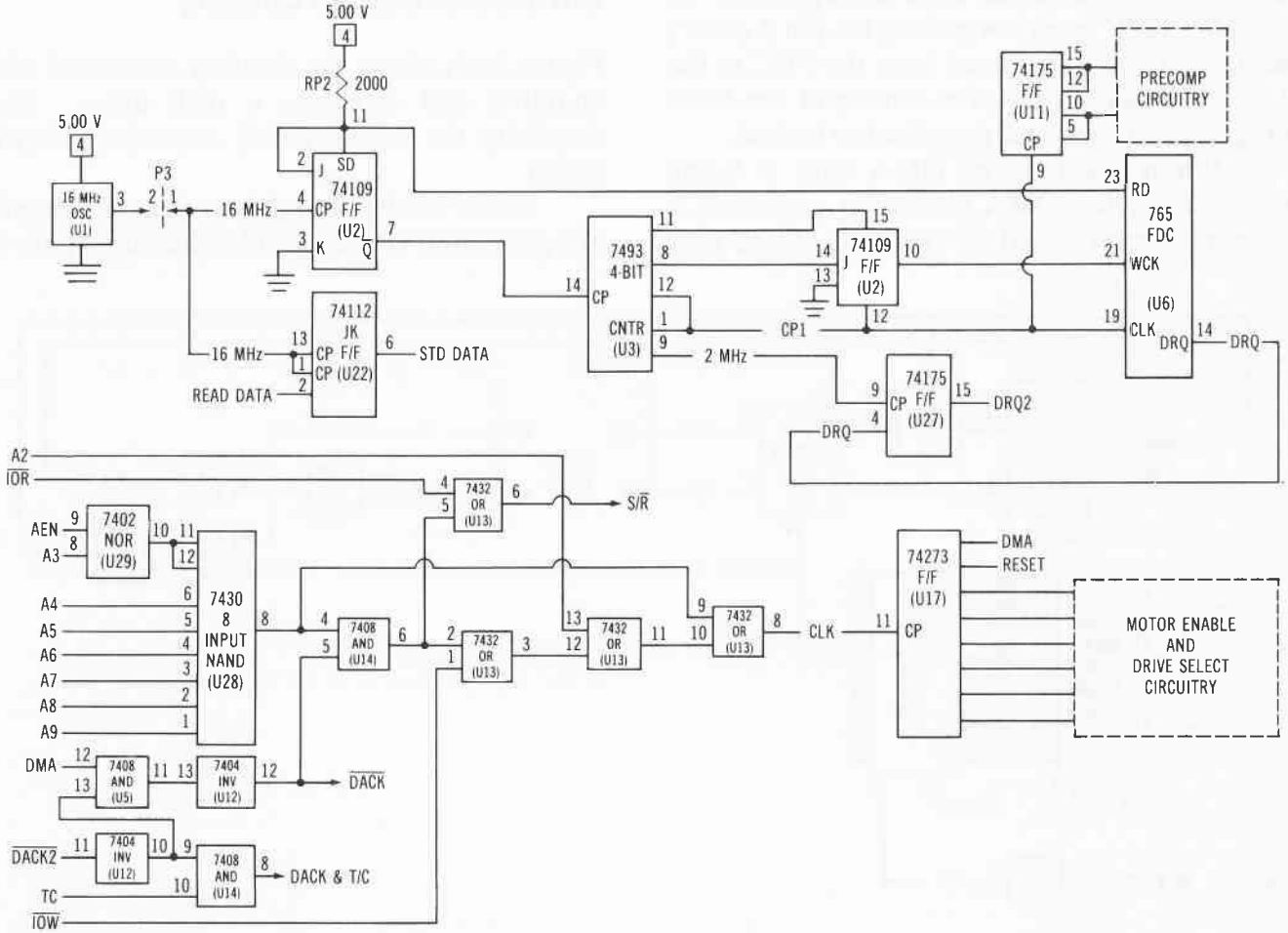


Fig. 2-99. Adapter board CLK, CPI, 16 MHz, and 2 MHz clock circuitry.

DMA Circuitry

Whenever I/O locations 3F4H or 3F5H are accessed, the clock signal CLK is generated and the internal registers of U6 are read. When U6 is ready to transfer a byte of data to or from the floppy disk, U6 pulls pin 14 (DRQ) high causing 3487 driver U7 to produce a DMA request signal (DRQ2) to pin B22 of P5, as shown in Fig. 2-100.

As discussed earlier in this chapter, DMA action occurs on the system board. The CPU relinquishes bus control to the 8237 DMA controller and it communicates with the drive adapter board to cause data transfer. The DMA controller generates an active low DMA acknowledge signal (DACK2*) that is inverted by U12 and ANDed with DMA from U17 to produce an active low DACK* signal for pin 15 of U6. Pin 15 active low disables U6 output pin 14 and selects the data register in U6 as the source, or destination, of data bus DD0 through DD7. In addition IOR* goes low pulling U6 pin 2 (RD*) so data can be transferred from the FDC to the I/O data bus. If a write command has been programmed, WR* will go active low instead.

When the byte of DMA data is being transferred, the DMA controller generates a terminal count signal (TC) that is ANDed with the now high DACK2* signal to produce a DACK&TC signal for U6 input pin 16. This signal indicates the end of a DMA transfer. This causes an interrupt signal to occur on U6 output pin 18 signifying the beginning of the FDC result phase. Pin 18 connects to pin 1 of 3487 driver U7. Under control of DMA on pin 4, U7 passes the interrupt signal out pin 2 as IRQ6 and through connector P5 onto the system board and into the programmable interrupt controller (PIC). The PIC, in turn, generates an interrupt to the 8088 CPU and an interrupt handler routine is accessed. When the first of the seven bytes of data is read out of the FDC during this result phase, the pin 18 interrupt signal is reset. Any data sheet on the 765 FDC part will describe the data bus contents during the various phases of controller operation.

Drive Motor Control Circuitry

Figure 2-101 shows the circuitry associated with enabling and selecting a disk drive. For simplicity, the drive A and B control circuitry is shown.

Motor enable and drive select commands are generated via the DMA data bus inputs to

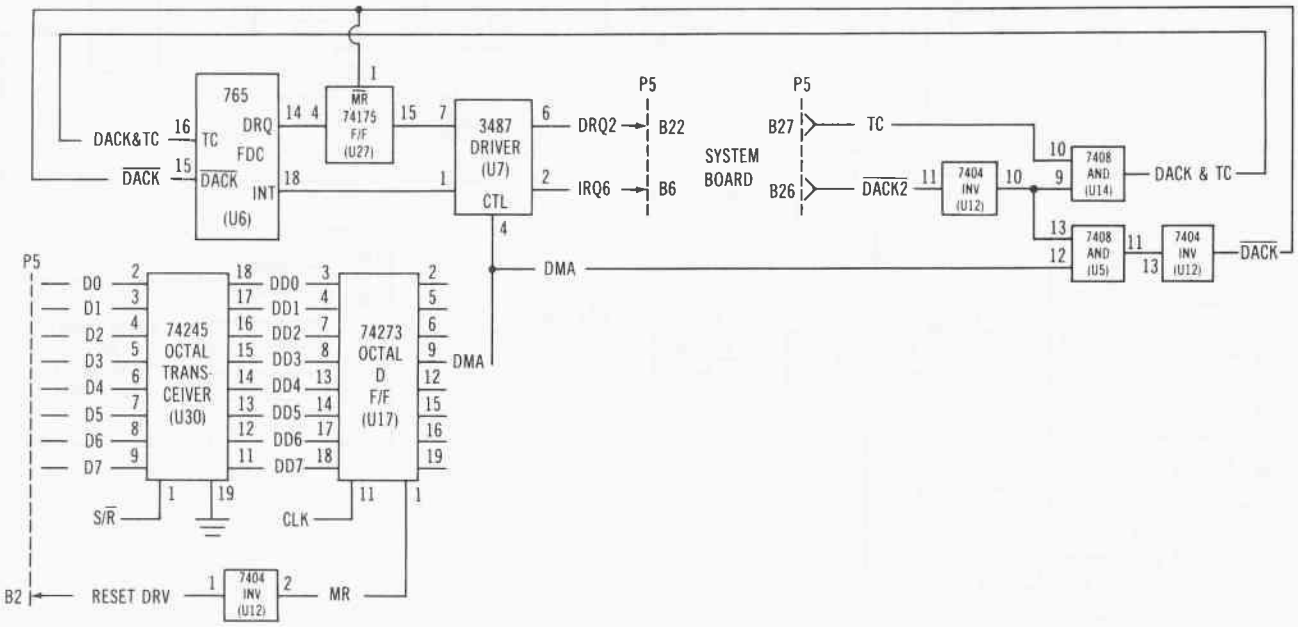


Fig. 2-100. Disk drive adapter DMA circuitry.

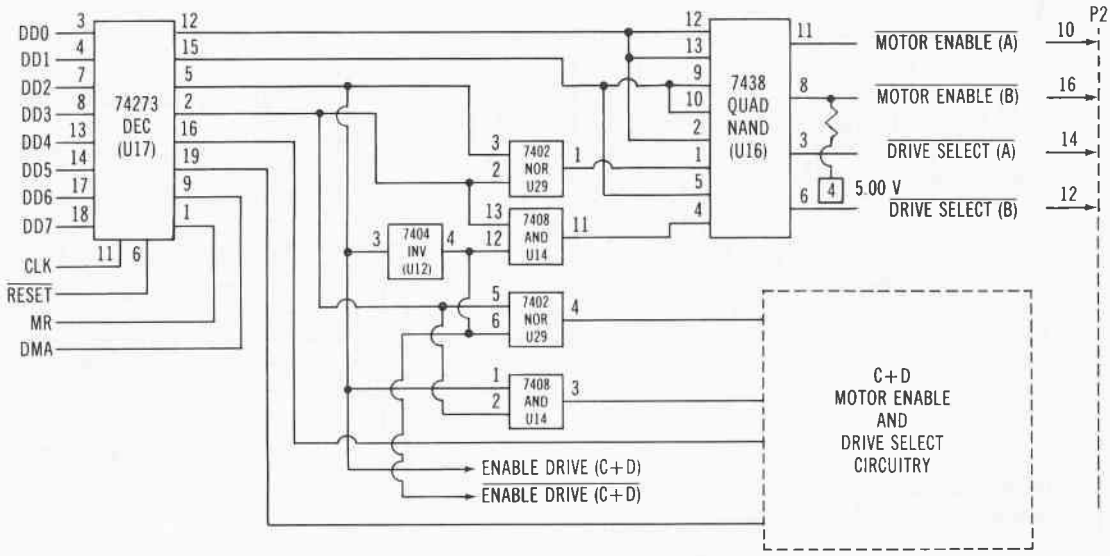


Fig. 2-101. Drive motor control circuitry.

74LS273 octal D flip-flop U17. The outputs from pins 2, 5, 12, 15, 16, and 19 are applied to a logic circuit of an inverter, a NOR gate, an AND gate, and a 74LS38 Quad NAND gate U16 to produce active low signals (MOTOR ENABLE A)*, (MOTOR ENABLE B)*, (DRIVE SELECT A)*, AND (DRIVE SELECT B)* at the output of U16.

The pin 5 output of U17 is also used as an active high ENABLE DRIVE C&D signal that is applied to the pin 3 input of 74LS04 inverter U12 and the pin 1 output enable of 74LS240 octal tristate buffer U16 in the drive to FDC read circuitry (described later). Inverter U12 complements this enable signal to produce (ENABLE DRIVE C&D)*. This signal is applied to another output enable pin (pin 19) of buffer U16.

Disk Drive Write Circuitry

Figure 2-102 is a block diagram of the drive write circuitry found on the disk drive adapter board. This circuitry generates the composite clock/data bit stream that becomes (WRITE DATA A&B)* and passes out over pin 22 of P2. It also generates a direction signal (DIR A&B)* out pin 18 of P2, a select head 0/1 signal (SELECT HEAD 1 A&B)* on P2 pin 32, and a step pulse out pin 20 of the same connector.

The combination of 75LS153 dual 4-line to 1-line multiplexer U10 and 74LS175 quad D flip-flop U11 generate a MFM serial clock/data stream that is output from U11 pin 2 to both the A&B and C&D drive circuitry. This signal is input to pins 1 and 2 of 74LS38 quad NAND gate U9 to produce an active low output signal (WRITE DATA A&B)* on pin 3. During the command phase of 765 operation, a set of nine bytes are sent to the FDC. This information includes the drive to select, the cylinder address, the head to select, the sector address, the sector size, the final sector number of the current track, the gap length between sectors (excluding the VCO synchronization field), and the effective sector size, so the FDC can write additional zeroes in case the write command completes before the end of a sector has been reached.

The execution phase begins and FDC U6 loads the head (if not already loaded), waits the specified 35 millisecond head load time, and begins reading sector ID fields. When a match is made between the requested sector address and the sector address read from the disk, the FDC reads data a byte at a time from the 8088 CPU and outputs the data to the magnetic head. The drive head writes this information into the data field area for that sector on the disk. When the data field is loaded, the 765 FDC computes a CRC value on the data and writes two CRC bytes

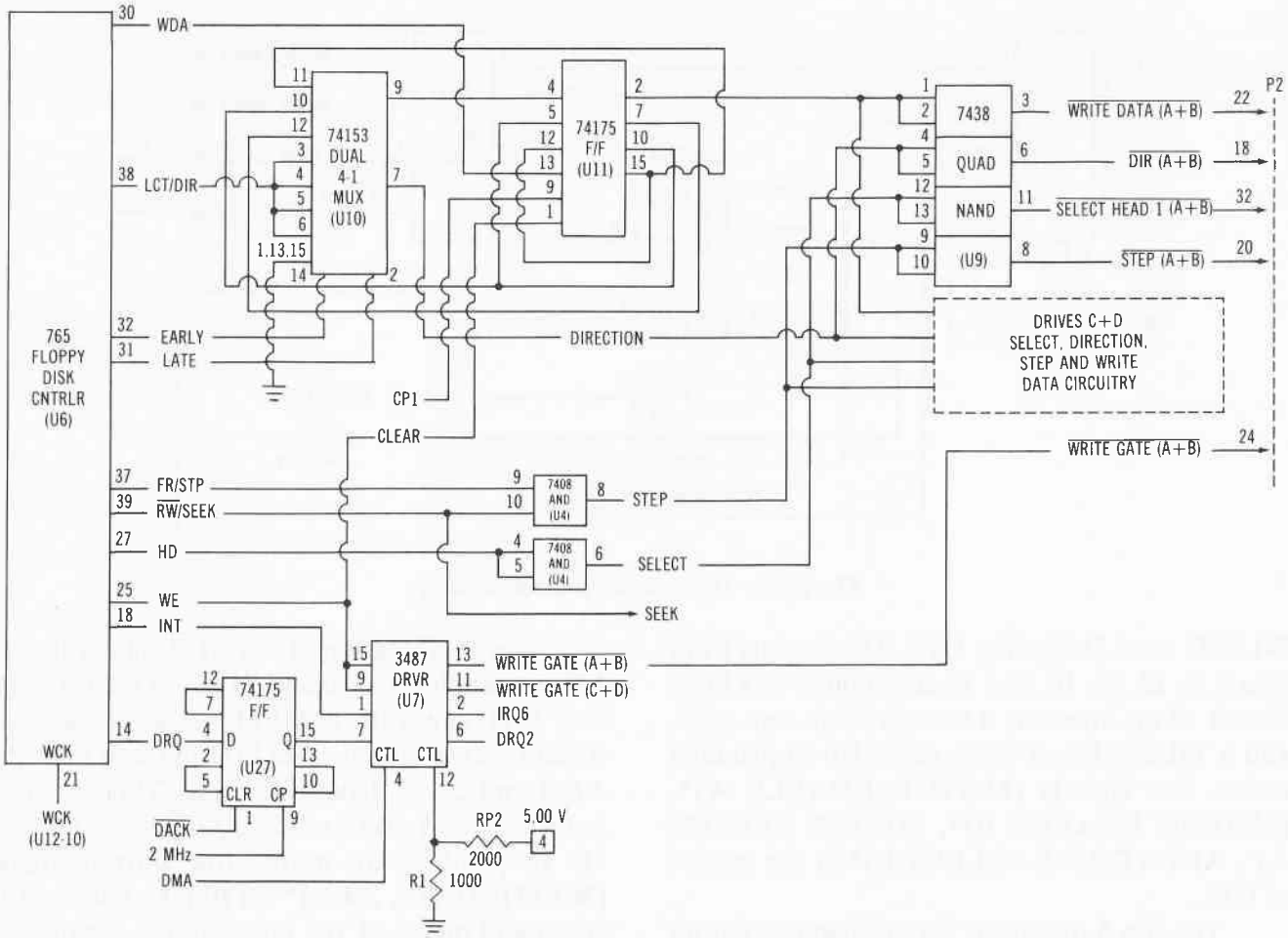


Fig. 2-102. Disk drive adapter write circuitry.

at the end of the data field. The FDC reads the ID field of the sector and compares the CRC bytes. If an error is detected, an error flag bit (bit 5) is set in the status register and the write data command terminates. Data transfers between the 8088 CPU and the FDC occur every 15 microseconds.

During the seek portion of the write cycle, FDC pin 38 outputs a logic value that determines the direction of head movement when a step pulse is received. A high causes the head to move (step) in; a low causes head movement to step out. When the write begins, pin 38 is used to lower the write current on the inner tracks of the disk.

FDC pins 31 and 32 are used by the precompensation circuitry of 74LS153 U10 and 74LS175 U11 to time-position clock and data pulses in the write data stream so that later playback will be able to properly separate out the clock and data from the signal stream. Preshift cancels predictable playback data shifts when MFM recorded data is retrieved from a floppy disk. A write clock signal (WCK) is input on pin 21 of 765 U6 (see also Fig. 2-97) to produce a 250 nanosecond clock signal that occurs at a 1 MHz rate. During a disk write operation, U6 specifies that a data bit should occur early, normal, or late relative to the write clock pulse. The early and late signals out pins 32 and 31 respectively are used to preshift the bit 250 nanoseconds before or after the write clock transition. If both output signals are low, no preshift occurs.

During the write mode, U6 pin 37 is also active. At the beginning of the write command, a fault reset pulse is issued to reset a fault flip-flop in the disk drive. When seek begins, pin 37 outputs step pulses that move the drive read write head to another cylinder track. During

seek, pin 39 is high. The signals out pins 37 and 39 are ANDed by 74LS08 U4 to produce a predetermined number of step pulses during the seek cycle. These step pulses are passed through 74LS38 NAND U9 to produce active low (STEP A&B)* pulses.

FDC pin 27 is a signal that determines which head is to be selected. When U6 output pin 27 is high, read/write head 1 is selected. This signal is inverted in the 74LS38 NAND U9 to produce an active low select head 1 signal (SELECT HEAD 1 A&B)* out pin 11 of U9.

Pin 25 on U6 is a write enable output that is applied to a 3487 driver U7 to generate active low (WRITE GATE A&B)* and (WRITE GATE C&D)* signals out pins 13 and 11 respectively. These active low signals enable write current in the selected read/write head. Bringing these lines high disables the write current in the head.

Disk Drive Read Circuitry

Figure 2-103 shows the disk drive to adapter board input circuitry. Connector P2 brings composite read/clock data, write protect status, index information, and track 0 information from drives A and B into the 74LS240 octal inverter tristate buffer U18. Connector J1 handles the same signals for drives C and D.

Regardless of which drive inputs the data, four outputs from U18 provide composite read data, write protect, index, and track 0 information into the board circuitry. Buffer driver U18 pins 9 and 12 connect to provide READ DATA from any of four drives to the phase-lock loop clock and data recovery circuitry.

Output pins 5 and 16 combine to provide a logic signal representing the condition of write protection on the selected drive's disk to pin 34 of U6. If a write-protected disk is mounted in the selected drive, WRITE PROTECT goes high. This input signal sets bit 6 of status register 3 in U6.

U18 pins 3 and 18 combine to form an INDEX signal that is passed into pin 17 of U6. When active high, INDEX indicates that the head has just detected the beginning of a new disk track.

Finally, U18 pins 7 and 14 are combined to produce TRACK 0. During a seek mode operation, the detection of TRACK 0 causes this

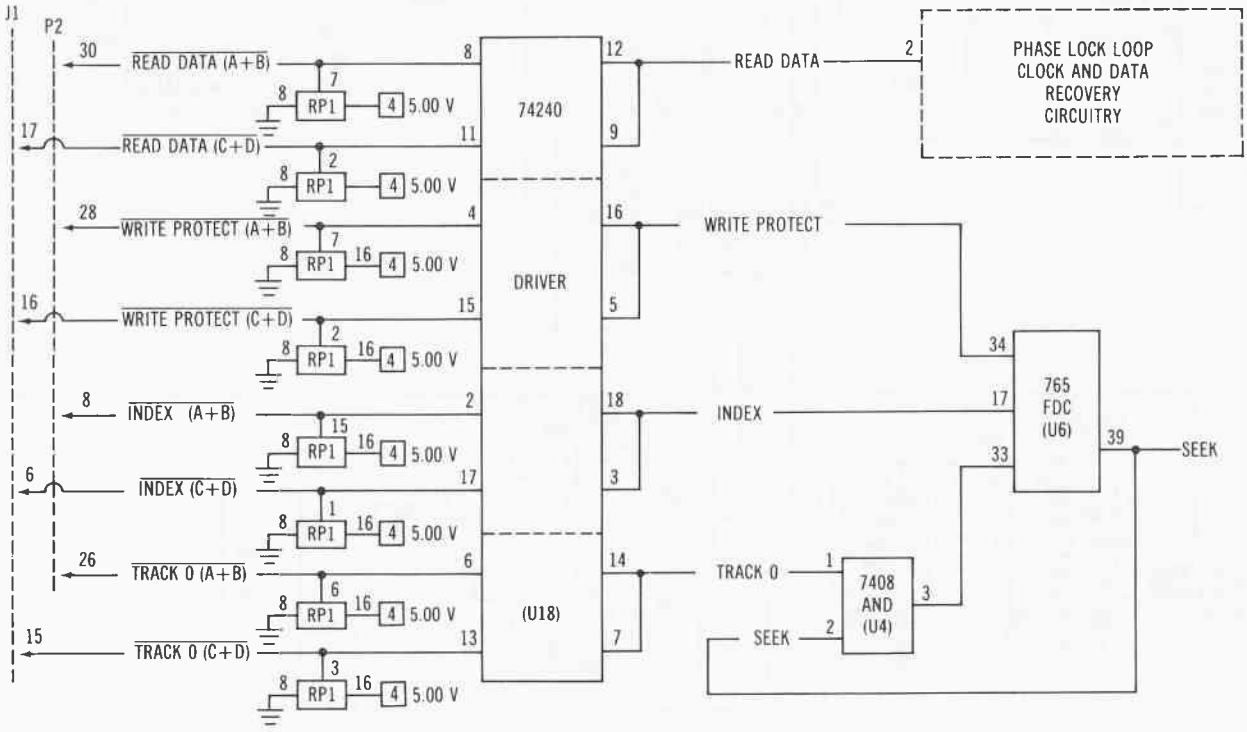


Fig. 2-103. Disk drive to adapter board input circuitry.

signal to go high. This sets bit 4 in FDC U6 status register 3.

Data Recovery Circuitry

During a read operation, the composite clock/data bit stream is buffered through U18 to a phase lock loop clock and data recovery circuitry, as shown in Fig. 2-103. Figure 2-104 expands the READ DATA handling circuitry to show the phase lock loop (PLL) pulse shaper, phase comparator, PLL oscillator, and FDC input read data and data window circuitry. The phase lock loop constantly analyzes the frequency of an input signal and locks another oscillator to the input frequency.

Composite clock/data bits are clocked into 74LS112 JK flip-flop U22 by a 16 MHz clock signal on pins 1 and 13. The FDC pin 24 output VCO SYNC is used to preset the READ DATA part of the dual flip-flop pair to the high state (output pin 6 held low) whenever VCO SYNC goes low. When VCO SYNC is low, it inhibits the voltage controlled oscillator in the phase-locked loop causing it to idle at its center frequency. After the read/write head has been loaded and the 35 nanosecond head load time has elapsed, VCO SYNC goes high enabling the pulse shaping and PLL circuitry. This circuitry is disabled in the gap between the ID field and the data field and in the gap after the data field before the next sector ID field to blank out erroneous data noise from head write current

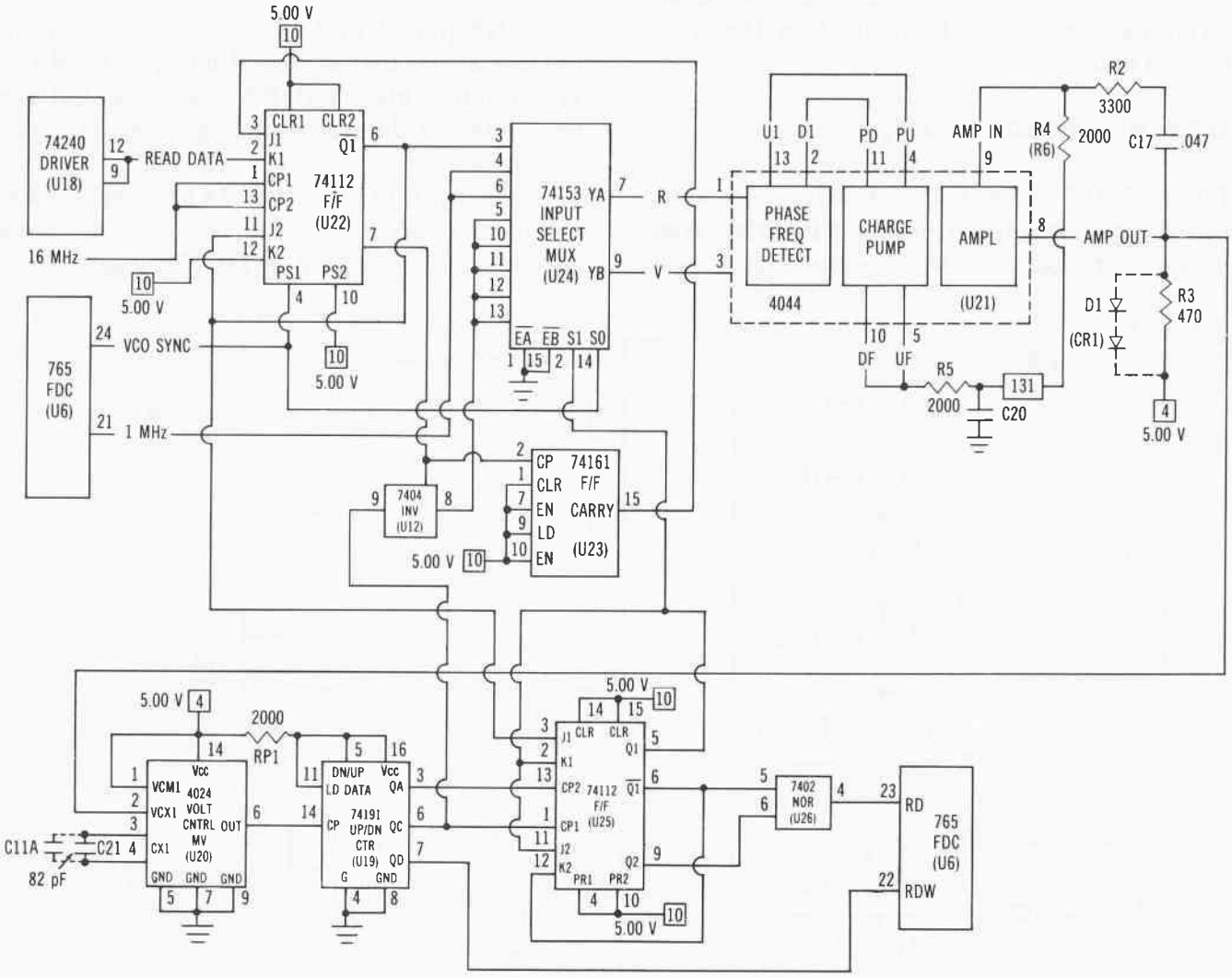


Fig. 2-104. Phase lock loop clock and data recovery circuitry.

turn on and off during gap time. VCO SYNC also serves as one of the two select signals for 74LS153 input select multiplexer U24.

A 4044 phase frequency detector (U21) is used as a phase-locked loop to compare the frequency of input data from the disk drive with the frequency of a 4024 voltage controlled multivibrator U20 functioning as a variable local oscillator. PLL U21 determines the clock and data bit positions in the READ DATA signal by sampling each bit. The phase relationship between a data bit and the PLL generated data window is constantly fed back to adjust the position of the data window causing U21 to track the READ DATA frequency changes and reproduce the same information that was sent from the system board to the drive electronics for storage.

In Fig. 2-104, 74LS112 U22 and 74LS153 serve to shape the data pulses before presenting to the input (pin 1) of PLL U21. At the output of U21 is an analog filter circuit that produces an error voltage from the PLL pump up (PU) and pump down (PD) signals derived from the phase difference between the PLL local oscillator U20 output and the READ DATA input.

The frequency difference AMP OUT of U21 is passed to pin 2 of 4024 voltage controlled oscillator (multivibrator) U20 to raise or lower its frequency so it better matches that of the READ DATA input signal. PLL U21 synchronizes the oscillator frequency of U20 to that of the READ DATA input during the reading of the "all zeroes" synchronization field off the track of the floppy disk. This field precedes the ID field and the data field.

Input select multiplexer U24 generates reference (R) and variable (V) inputs to 4044 phase frequency detector U21. The phase difference between READ DATA on the R (pin 1) input and the VCO U20 signal on the V (pin 3) input is compared in the phase frequency detect portion of U21. Loop "lock-on" occurs when both U1 (pin 13) and D1 (pin 2) remain high. This occurs when all the negative transitions of READ DATA R and V coincide. This circuit only responds to transitions and is independent of waveform duty cycle or amplitude variations. The outputs U1 and D1 are recycled back into pins 4 and 11 of the charge pump section as pump down (PD) or pump up (PU) error signals. If the READ DATA pulse occurs early compared with the VCO U20 variable pulse, the PU duration will be shorter than the PD duration. Likewise, if a READ DATA pulse occurs late, PU is held high longer than PD. The difference between the PU active time and the PD active time represents the difference between the READ DATA bit rate and the PLL clock rate. If PD is active longer than PU, the input bit rate is slower than the PLL clock.

The PU and PD signals into the charging pump produce the error voltages down frequency (DF) on pin 10 and up frequency (UF) on pin 5. These signals are applied to an analog RC filter network including an amplifier circuit inside U21. The DF and UF outputs from U21 are filtered and amplified to produce an error voltage out U21 pin 8. The relationship of PU and PD active times modifies the PLL oscillator time constant via the filter output to produce a VCO error voltage. This error voltage is passed to input pin 2 of the 4024 voltage controlled multivibrator oscillator U20 and causes the pin 6 output frequency to vary relative to the amount of error voltage.

Capacitor C21 (82 pF) is connected across pins 3 and 4 of U20 to control the operating frequency range of this oscillator. In some designs, a second 82 pF capacitor (U11A) is connected in parallel with C21.

The pin 6 output of U20 is connected to input pin 14 of 74LS191 up/down counter U19. Pin 5 (U*/D) is strapped high through a 2K resistor RP1 causing U19 to operate in the countdown mode. Output Qa (pin 3) functions as a pin 13 clock input for 74LS112 flip-flop U25. Pin 6 clocks another part of U25 to enable a composite READ DATA signal from output pin 6 of the 74LS112 input PLL pulse shaper flip-flop U22 in the upper left of Fig. 2-104 to clock out the Q1* (pin 6) and Q2 (pin 9) lines of U25 through 74LS02 NOR gate U26 into pin 23 of U6.

Pin 7 of U19 outputs a data window signal that is passed to pin 22 of 765 FDC U6. This signal (RDW) is used to isolate the data bits contained in the READ DATA composite input signal entering pin 23 of U6. After isolation of the data bits from the READ DATA bit stream, U6 assembles the data bits into 8-bit bytes for transfer to the 8088 CPU or system board memory.

SUMMARY

In this chapter, a detailed analysis was conducted into the operation of the IBM PC system board, keyboard, display, and disk drives. Knowledge to this level is required by technicians who desire to apply an intimate understanding of the PC electronics to achieve quick and correct troubleshooting of IBM PC computer problems.

3

Troubleshooting Techniques

Verifying a real problem, analyzing symptoms, and isolating and correcting a failure in the IBM PC can be a stress-filled hassle, or an enjoyable, rewarding experience depending on your understanding of the machine and your ability to properly troubleshoot computer circuitry. Whether you're a hobbyist, or a technician, this chapter will guide you through the troubleshooting process and provide you with analytical tools and tips that will help you analyze a problem, identify the failed part, and step toward the correct repair.

INTRODUCTION TO TROUBLESHOOTING

Imagine for a moment that you're in the midst of printing a lengthy analysis report when suddenly the printer halts, the screen display goes blank and your IBM PC ceases to function. What do you do? What failed?

This chapter is devoted to a subject we often wish we could pass off or ignore–trouble. Trouble is like a flat tire: no one wants one, but when it occurs we all wish we could fix it quickly and get the experience behind us. Knowledge and action are required to overcome trouble.

Integrated circuit technology is advancing rapidly. Logic gates on tiny chips of silicon are getting smaller and faster. This has been welcomed by all, but coming with these advances in microelectronics are more challenges to overcome in determining whether a chip, board, or computer system is functioning correctly and has been properly maintained. Faults can occur that are difficult to locate.

A fault is any physical condition that causes incorrect output when a circuit is exercised to perform a function. Faults can be classified as static or dynamic. Static failures include the stuck-at problems associated with open or short data paths in circuitry. These failures are typically catastrophic causing a system operation to end. Shorts can be described as electrical conduction in the wrong place. Shorts are typically caused by mechanical failure in a device or a solder bridge during improper repair. Opens are cases of no electrical conduction when it should be present. Electrically open inputs can affect the switching speed of a device. They can also degrade the noise immunity of the component. Other catastrophic faults include the wrong component installed on a board, an improperly installed component, a missing component, and dead or partially dead devices.

Dynamic failures include time-dependent errors such as the loss of signal quality causing a circuit output to reach steady state too late to be properly used by another part of the system. The symptoms of dynamic faults include devices operating too slow. This failure is seen in setup and hold problems, data and addressing problems, machine cycle-time instability, and interactive problems between components. In the logic gate, dynamic faults are seen in propagation delay problems–time delay in getting a signal from the input to the output. Flip-flops typically experience dynamic faults in their setup and hold ability to capture and hold data after the inputs have dissipated. Dynamic faults in memory occur in the data/address relationships where timing problems occur between the occurrence of valid data and address information. Even the 8088 central processing unit can experience dynamic faults in the cycle-time stability of the microinstruction function cycle. Component dynamic failures are more difficult to find than the static catastrophic faults. Locating static and dynamic faults will be covered in this chapter.

The most effective way to locate a failure in the IBM PC is to think the problem through just as the machine operates–logically. Imagine the computer system as a human body. The timing and the timing circuitry represent the heart. The CPU and related ciruitry are like the brain. Without the heart and brain, nothing works in the body. The keyboard and drives represent the eyes and ears. The display and printer act like the mouth. By viewing the computer system as a functioning body system, you can quickly determine which area is not working properly and home in on the malfunctioning part. Understand what should happen and compare the "shoulds," one by one, with what is really happening.

There are typically two ways to analyze electronic circuit failures: classical troubleshooting which incorporates localizing and isolating a failure using deductive reasoning and mental intuition; and brute force troubleshooting which uses flow charts and replacement of all suspected components. Both of these techniques will be addressed in this chapter.

CLASSICAL STEPS TO SUCCESSFUL TROUBLESHOOTING

Solving computer system problems requires application of the deductive technique called "troubleshooting". Effective and efficient troubleshooting requires gathering clues and applying deductive reasoning to isolate the problem. Once you know the cause of the problem, you can follow a process of analyzing, testing, and substituting good components for each suspected bad component to find the particular part that has failed.

The use of special test equipment such as logic probes, logic clips, digital multimeters, oscilloscopes, and logic analyzers are the technician's tools-of-the-trade to help speed the process. Good, deductive reasoning is used to isolate a failure to a particular group of chips, then circuit analysis is used to reduce the problem to a specific component. When a suspected circuit network is found, there are two ways to test the board: in-circuit testing and functional testing.

In-circuit testing treats the PC system board as a collection of parts. Testing is accomplished on each individual part as though it were all alone. In-circuit testing relies on the ability of the tester to isolate and test the board components separately. Test patterns are automatically applied to the inputs to each component on a board, and the responses are measured directly at the outputs. This is a good diagnostic fault isolation technique for locating catastrophic faults, but if not done properly, in-circuit testing can cause catastrophic failure or degradation of component performance. Improper application of this technique can produce high current density, local hot spots, and excessive voltages in certain parts of the circuitry. These problems can be minimized using dedicated layout strategies and design-for-test techniques.

A specific form of in-circuit testing in which every electrical network on the PC system board is accessed at the same time is the bed of nails technique. A bed of nails fixture is typically a box with rubber gaskets that form a tight seal with the printed-circuit board being tested. It has

hundreds (or thousands) of prongs (nails) with spring-loaded probe tips. When the circuit board is closed inside the vacuum of the bed of nails box, the spring-loaded probe tips make solid contact with the solder side of the board. While this is a very useful test technique for printed-circuit boards, the development of the nail configuration, construction of the box, and generation of the test program that monitors all test points is very expensive and generally out of reach of most service centers. A subset of in-circuit testing is accomplished by most service technicians using logic probes, pulsers, and other test tools that probe various parts of the board circuitry.

Functional testing treats the PC system board as a single functioning entity. This test technique evaluates the board in an environment that closely emulates the system for which it was targeted. A subset of functional testing is found in the short programs that can be written to exercise certain functions in the system. When these programs are executed, test equipment monitor specific test points to determine the proper (or improper) operation at those nodes.

Another form of functional testing occurs when test patterns are applied at the inputs to a circuit and the outputs are observed and measured. The expected results are derived by simulating the test patterns against a good machine which serves as a model. This was the technique used to produce the voltages and waveforms noted on the COMPUTERFACTS schematics. When the same circuit stimulation criteria are used on a suspected bad circuit, the measured results can be compared with the results derived from the known good model.

The key here is that the test node must be observable and vitally involved in the function being done. A related requirement is that the measuring equipment must not introduce adverse reaction by the circuit under test. It must not load down or change the signal content at the test point.

For example, the output of a gate can be considered the output of a transistor configuration. The input of a gate can be considered the input to a transistor configuration. Transistor inputs characteristically have very high impedance; the outputs have very low impedance. Touching a point on the circuit board causes the tester to source or sink current. Most of the current drives toward the points of low impedance. This causes the output transistor to be back driven producing heat in the chip. The temperature of the output junction increases. As long as the thermal threshold of the transistor is not violated, no component damage occurs. The accepted temperature threshold for silicon transistors is typically 120 degrees Celsius. Therefore, probe tests in an active circuit should not be overused.

In general, there are some favorable steps that you can follow to achieve successful computer troubleshooting and repair.

1. Don't panic.
2. Observe the conditions.
3. Use your senses.
4. Retry.
5. Document.
6. Assume one problem.
7. Use correct service data.
8. Use the right test equipment.
9. Diagnose to a section (fault identification).
10. Localize to a stage (fault localization).
11. Isolate to a failed part (fault isolation).
12. Repair.
13. Test and verify.

The following pages discuss the steps to troubleshooting success in detail.

Every computer is composed of functional sections, as shown in Fig. 3-1. Any of these sections can fail.

When something functionally goes wrong in the computer, the first step is to determine whether the trouble results from an actual failure or from a loose connection or human error. To do this, you need to understand how the IBM PC works and how it interacts with the other parts of the system. Chapter 2 was written to fill this requirement.

Once you're convinced a true component failure has occurred, the next step is to determine

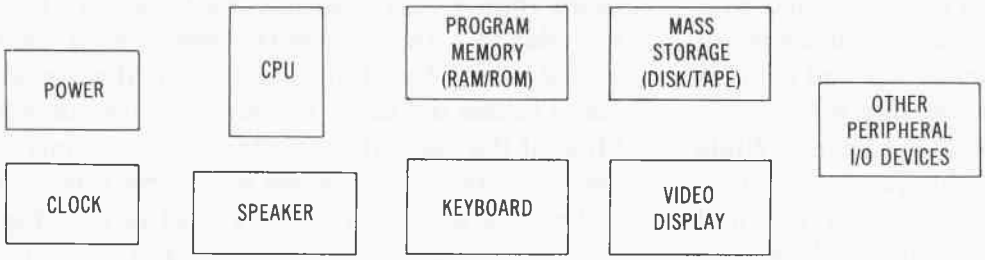


Fig. 3-1. The functional units of the IBM personal computer.

which functional section of the system is not operating–disk drive, keyboard, display, or some other part. To do this, break each section into stages and trace the trouble to a circuit stage within the section. If a display isn't working, for example, the problem could be in the display monitor itself, in the video cable, or in the video circuitry of the computer. Each of these can be considered a stage of the video display functional section. Then, analyze the circuit to isolate the failed part.

When troubleshooting a computer you must discipline yourself to check that the power is what you need it to be at that time (usually off). Make it a practice to always place your hand over the power switch whenever you first start thinking about doing something inside the computer.

Visual Inspection

There are specific steps you should take when troubleshooting an IBM PC. First, search out all the symptoms, the clues, that point toward the location of the failure. Make a visual and operational check of everything that is active during normal operation of the function that is failing. Look for misplaced or unconnected cables, power and other switches incorrectly set, disk drive doors inadvertently left open, unplugged wall sockets and bad disks. Look for anything that appears out of place.

Cleaning Connections and Inside the Chassis

Turn off the computer and clean all the edge connectors on the plug-in cards. Reseat the associated cables, making sure to look for bent pins. Examine the system board and related interface cards for discolored components or loose debris. You'd be amazed at what has been found inside computers, printers, and disk drives. During analysis, we've found pieces of bread, cigarette ashes, coffee spills on the keyboard, and even sticky soda pop all over the motherboard. Inside drives we've found everything from pencils to carrots to strange creatures. In one drive that was being repaired for a school district, we found a dead mouse! Don't be surprised by the things that somehow find their way into these machines.

After checking the internals of the computer, disk drive, and any other associated equipment, close the system up and reboot. Check to see if the same failure occurs. If it does, shift to symptom analysis.

Symptom Analysis

In this step in the troubleshooting process you examine symptoms. Carefully evaluate the problem to determine what area of the system is failing. If the screen on the display console is black and shows no sign of video life, then check the monitor and video circuitry and interconnecting cables. If the program locks up in the middle of an operation, the failure is most likely in the CPU, ROM, RAM, or related circuitry. If the drive doesn't boot, then the drive, the drive circuitry, the disk drive adapter board and the related interface cables become suspect.

When checking symptoms, remember to check for all symptoms, not just ones that seem directly related to the problem. Many times there are other clues available.

Diagnose to a Section

After you've considered all the symptoms, narrow the failure to the section in which the

problem seems to be centralized. For example, lets say you have a PC in front of you whose disk won't boot in the disk drive. You completed the visual check and found nothing, so you analyze the failure symptoms. You notice that when power is applied to the system the computer turns on, and the self-diagnostic runs and passes.

Keep in mind that, even though the diagnostics passes, there can still be a problem in the circuitry that was tested. The PC diagnostic does not test all types of stuck at or improper signal conditions. In our example, the diagnostics run fine but when the drive tries to boot it just locks up and doesn't do anything and no failure code is displayed on the screen.

You turn the computer off and, several seconds later back on, closely watching the action of drive A. When the BIOS reaches the step to boot the disk, the drive light turns on and you can hear head movement inside. However, about 40 seconds later, the drive motor stops. The drive light is on, but no activity occurs. A curious fact. You try another known good disk. Same results. You swap disk drives and retest. Again, no change. You clean and reseat the drive adapter board. You also clean and reconnect the cable from the drive to the adapter board. You retest. The problem remains.

In our example, you remove the disk from the drive and power down and back up trying to boot the system up in BASIC. It does. Writing a short program verifies that the software works fine.

Now that the system has been proven to work without the drives, you conclude that the CPU, the system board, the screen, and the keyboard sections work properly. The problem has been reduced to the disk drive portion of the system circuitry. Closing in on the failure, you decide that you have a choice to make. If you have spare cables and adapter boards you can swap out one at a time until the problem goes away. However, if you don't have spares, you must investigate further. Look at the magnetic head inside the failing drive as you reapply power to the system. It should immediately move to the "home" position furthest from the center of the hub. This is the location of Track 0. Disk unique information such as where the directory can be found is located on this track. In this example, the head goes to the home position. Then the head moves across the disk just like it was reading data off a track cylinder, but 40 seconds later head movement stops with the head positioned half way across the disk. It's now time to break out the scope to take technical measurements.

You check the basic power and timing circuitry. Power and a good ground are present at the drive and adapter board. You check the speed and tracking of the malfunctioning drive. (Chapter 4 Preliminary Service Checks provides a detailed description of speed and tracking adjustments.) These are verified within specification so you shift into detailed troubleshooting.

Localize to a Stage

You check for open or shorted pins in the cable. Then you check the disk drive controller circuitry on the adapter board. The problem localizes to a circuit stage on the adapter board. Normally, you would have found the problem by now. (Chapter 5 covers detailed troubleshooting.)

Isolate to a Failed Part

You isolate and locate a failed part on the adapter board. Replacing the part, you verify proper system operation has been restored.

The specific problem described in this example is not important. The value of the preceding analysis is the troubleshooting steps that have been described. The first step is a visual check (including swapping disks with a known good disk). Next, the problem is reduced to a sub-system by checking symptoms. Then a preliminary check is conducted on the circuitry in question. This is followed by specific circuit analysis to isolate and identify the failed component.

UNDERSTANDING HOW COMPONENTS FAIL

While the use of troubleshooting equipment makes it easier to analyze and isolate different computer problems, many failures can be found using deductive reasoning and understanding. In fact, troubleshooting and repair can be relatively simple if you know how the system should operate and understand how electronic components fail.

Failures generally occur in the circuits that are used or stressed the most. These include the RAM and ROM memory chips, the 8088 CPU, and the input/output (I/O) chips between the motherboard and the peripherals. The CPU is a highly reliable device and seldom fails. Most failures involve the other chips. Except for the ROM chips which are programmed by IBM, most of these other chips are standard, off-the-shelf devices and are so common they've earned the nickname "jelly beans"–inexpensive, easy-to-replace products. They can be obtained at a low cost at almost any electronic store or from an IC distributor.

Integrated Circuits—Chips

A chip or integrated circuit is constructed out of silicon with some other tiny particles of metal (impurities) imbedded in specific positions in the silicon. By positioning the metals in certain ways, tiny transistors can be formed. Applying a voltage to specific locations on the chip allows the device to invert a voltage level (+5 volts, logic 1, to 0 volt, logic 0), and enable all sorts of logic gates (AND, NAND, OR, NOR, and so forth) to function. It turns out that these chips can be made with silicon/metal junctions so tiny that today thousands of transistors can be placed on one chip. A memory chip the size of a fingernail can hold over 470,000 transistors.

The problem for chip manufacturers is how to get voltages and signals in and off such a tiny chip. Very thin wires are used as inputs and outputs to the chip. These wires are glued or bonded to tiny pads on the chip. The other end of each wire is bonded to a larger pad on a supporting material (the big part of what we call the integrated circuit, as shown in Fig. 3-2).

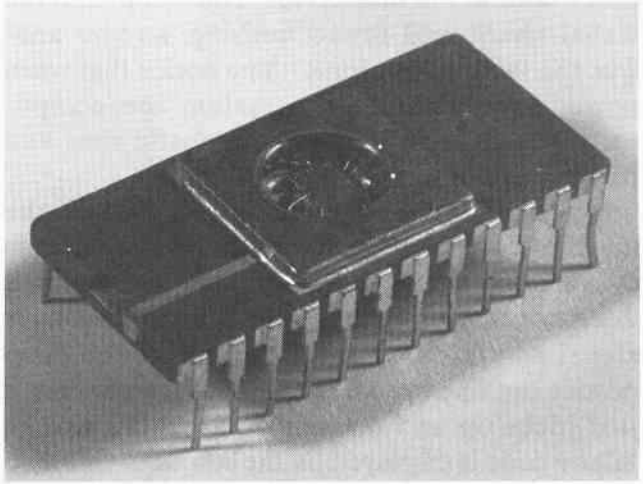


Fig. 3-2. Photo shows the chip pad to package interconnect leads.

The supporting structure includes the pins we plug into the sockets on our printed-circuit boards.

Integrated circuits are designed to operate for many hours, as shown in Table 3-1.

Table 3-1. Life Test Data for Various IC Technologies

Technology	Failures Per Device Hours
Bipolar	1 per 6 million
CMOS	1 per 55 million
NMOS	1 per 100 million
NMOS μPs	1 per 40 million
PMOS	1 per 10 million
PMOS μPs	1 per 35 million
8080 μP	1 per 16 million
4K SRAM	1 per 10 million

These tiny silicon and metal chips are placed in environments that put them under thermal stress during normal operation. Thermal stress affects those tiny strands of wire, or leads, going between the chip and the supporting structure including the large pins that are inserted into sockets. After a period of time, the thermal stress can cause the bonding of the wire lead to break away from the pad on the chip.

This disconnect causes an input or output to become an "open" circuit, and chip replacement is required. These interconnection failures are principally caused by electromigration. Silicon aluminum and copper aluminum were invented to replace the older pure aluminum interconnection wiring because they can tolerate higher current.

Another failure in these chips is caused by a phenomenon called "metal migration." The chip can be compared with an ocean of atoms. Some tiny particles of metal float about in this sea, migrating in directions perpendicular to electrical current flowing through the chip. Problems occur when these metal particles begin to collect in parts of the chip. If they concentrate in the middle of one of those microelectronic transistors, they cause the transistor to operate differently or not at all. If the resistance of these collected metals gets high enough, it causes the device to operate intermittently or to simply refuse to work. Since a transistor is part of a logic gate, the gate malfunctions and the output may become "stuck at 1" or "stuck at 0," no matter what the input signal is. Theoretically, a wearout failure won't occur until after several hundred years of use. We shorten the life span of our chips by placing them in high temperature, high voltage, or power cycling environments. These cause the devices to fail sooner.

Any event that acts to break down the oxide on a silicon chip can become a failure mechanism. Hot electron effects caused by injecting electrons into the oxide weakens the breakdown strength of the chip oxide. This also occurs over time (time-dependent breakdown).

In bipolar and MOS technologies, the most important failure mechanisms are metal corrosion and surface inversion. Half of the failed parts returned from the field exhibit these failure mechanisms. These problems can generally be attributed to poor assembly procedures and poor chip packages.

The second most important failure mechanism in ICs is caused by electrical overstress. Approximately one-fourth of the returned parts tested showed failures caused by operating the device in high temperature or high humidity. This was known to be a problem with MOS circuits, but now with the use of oxide isolation in bipolar circuits, these devices are also being adversely affected.

The remaining fourth of the failures found in returned parts showed physical malfunction caused by pinholes in the oxide and broken interconnections from electromigration.

In the manufacturing process, the major contributing aspect to failure are dust particles that cause mask and oxide defects. In fact, dust particles are believed responsible for 80 percent of all fab-line failures in today's MOS chips.

The I/O buffers have a higher failure rate than the internal logic cells because they experience higher currents and consume more power. On many bipolar parts, making them TTL compatible causes high power dissipation at the chip I/O. Other problems occur outside the chip, between the chip leads and the support structure pin leads, the device inputs or outputs.

These types of failures include: inputs or outputs shorted to ground, pins shorted to the +5 volt supply, pins shorted together, open pins, and connectors with intermittent defects. The most common IC trouble (assuming power is available) are opens or shorts to ground. Under normal use, chips finally fail with an input or output shorted to ground.

I/O pins and the bonding wires that connect the package pins to the pads on the IC die bonded inside the package have a higher failure rate than the devices on the chip because these interconnections are susceptible to environmental stress such as temperature variations, vibration, and improper handling and operation. Historically, only two percent of all IC failures occur in the silicon of the chip. Ninety-eight percent of all IC failures occur in the interconnects–the bonding wires and the leads coming off the package.

The classical stuck-at faults that came out of the diode-transistor logic era have been joined by nonclassical faults such as shorts between adjacent signal paths changing functions in an IC circuit. A stuck-open in a CMOS transmission gate or a reduction in transistor gain caused by ESD damage can cause slow gate switching.

Excess charge leakage or charge injection can cause slow independent change in a logic level. A high simply drifts down to a low.

Typical IC failure mechanisms are summarized in Table 3-2.

Table 3-2. Typical IC Failure Mechanisms

Chip Failure Mechanism	Package Failure Mechanism
Oxide faults	Broken wires
Oxide/junction contaminants	Lifted bonds
Diffusion defects	Gross deformation
Mechanical defects	Particulate contaminants
Metallization defects	Chip die separation
	Loss of hermeticity

Many failure mechanisms in ICs cause intermittent malfunction. These effects are typically caused by design specification margins that are too tight, fabrication line variations and contaminants, and random events such as alpha particle impact inside a package. Test methods that seek out classical stuck-at faults are not useful for chasing down intermittents.

Diodes and Transistors

The diodes and transistors on your computer's system board and adapter cards are made of solid material and act much alike. In fact, the transistor can be considered as partly constructed of two diodes.

Diodes are one-way valves for electric current, allowing current flow in only one direction. Diodes are usually made of either silicon or germanium. They are used in power supplies as rectifiers and in some circuits to maintain a constant voltage level. Other diodes are made of gallium arsenide and react by giving off light when biased in a certain way. These are called light emitting diodes or LEDs.

Transistors are used in various places in your computer circuitry as amplifiers or electronic switches.

Transistors and diodes fail by disconnecting inside, which causes an open or break in the circuitry, or by having their output short. Either kind of failure causes total loss of signal.

Diodes and transistors fail in the same ways and for the same reasons as chips, but chips fail more often than diodes or transistors. One reason is that there are many more tiny transistors on a chip the same size as a single (discrete) diode or transistor. This produces more heat and hence more thermal wear in the chip.

Capacitors

An understanding of the way a standard capacitor is constructed will aid in your understanding how these devices fail.

There are several types of capacitors on the PC system board and adapter cards. The capacitor is constructed of two separated plates. A voltage is placed across the plates and for a short instant, current flows across the gap. But soon electrons build up on one plate and cause the current flow to stop. The capacitor is then considered charged to some voltage potential. Capacitors are used to store charge and to filter unwanted signal spikes (sharp, quick peaks of voltage) to ground.

The electrolytic capacitor is constructed as shown in Fig. 3-3.

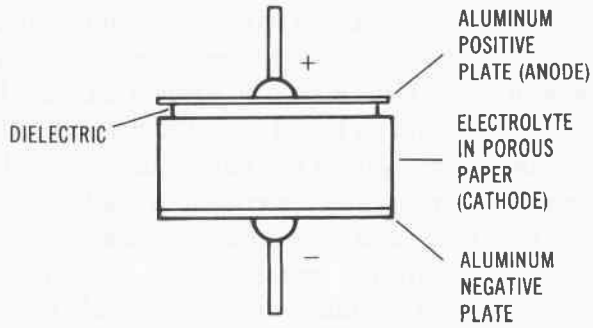


Fig. 3-3. A block diagram of the electrolytic capacitor.

Two aluminum foils or plates are separated by a layer of porous paper soaked with electrolyte, a conductive liquid. On one plate (the positive plate) a thin layer of aluminum oxide is deposited. This is called the dielectric. A

capacitor has an anode (the positive plate) and a cathode (the electrolyte). Electrons build up on one plate causing it to become so negative that it prevents further current flow (remember that electrons have a negative charge).

Another type of capacitor found in the PC is the "film" capacitor. It is constructed of alternating layers of aluminum foil and a plastic (usually polystyrene) insulation. The metal foil acts as the plates and the plastic insulation acts as the dielectric between the plates. Film capacitors are coated with epoxy and have tinned copper leads.

Capacitors open or short depending on the operating conditions and on their age. Capacitors fail when they short internally or when one of the leads disconnects, causing an open. Again there is a loss of signal.

Electrolytic capacitors are especially susceptible to the aging process. One effect of aging is drying out of the electrolyte insulator. The capacitance value increases, and circuit performance decreases. Finally the capacitance value drops dramatically as the plates fold toward each other, and shorting of the plates can occur.

Another kind of failure occurs when some of the dielectric oxide dissolves into the moist electrolyte, causing the thickness of the dielectric to shrink. This deforming usually occurs when the electrolytic capacitor sits for a long time without voltage applied. Here, the capacitance value increases but a high leakage of electrons occurs across the plates, making the capacitor useless.

The leads of the capacitor can physically detach from its plate causing an open in the circuit.

Also, the plates can short together when a large area of one plate is stripped of its dielectric oxide layer by the application of too much voltage.

Resistors

These current-limiting, voltage-dropping devices are quite reliable and should function properly for the life of your computer. However, the features that shorten the useful life of the chips also act to reduce the operational life of resistors. High temperatures, high voltage, and power cycling all affect the materials of which the resistors are made. These stresses cause breaks in the carbon, resistive paste, or resistive layers producing an open conduction path in the circuit. Excessively high voltages can produce electrical current so large that it actually chars resistors to burnt ash. This is rare, especially in a digital circuit where the highest voltage seen is 12 volts (usually 5 volts) and the currents are very tiny indeed (milliamperes).

Resistor failures are almost always associated with catastrophic failure of some other circuit component. Resistor failures, when they occur, are usually located in printer electronics rather than in the IBM personal computer. Like capacitors, however, resistors in the PC digital logic circuitry will seldom fail. A momentary power surge through the power supply can "fry" a capacitor on your PC system board. Excessive temperatures can also damage resistors.

Resistors can absorb too much current and actually bake in the circuit. The result is usually an open circuit with shorting during the "meltdown."

All the devices mentioned so far are solid state. These components are built to rigid specifications and are constructed of materials (metals, plastics, oxide, and so forth) whose electronic performance changes as the components age. We can accelerate this process by producing excessive voltages in the circuitry, or allowing the system to operate without adequate cooling. Severe temperatures or high voltages can cause the device and the circuit or system to behave strangely. Fortunately, IBM PC motherboards are not exposed to high voltages. But they can get hot (especially if you plug a lot of cards in the expansion slots), and this will affect the operation of the components.

Anytime a computer is operated, the circuit components (especially the chips) are subjected to thermal stress. First they heat up when the machine is energized. Then they cool down when the PC is turned off, and they heat up when we turn the machine on again. This heating up and cooling off weakens the ICs and eventually

causes failure. Thermal stress can cause a break in the connection of a wire leading from inside the chip to a pin, producing an "open" circuit, which requires chip replacement.

Even if no break in the chip or lead connection occurs, lengthy exposure to high voltages or temperatures can change the operating characteristics of a device. When the performance of these devices falls far enough below specs, the system starts to fail. A chip may work intermittently or simply refuse to work at all. An output can become stuck at 1 or stuck at 0, no matter what input signal is applied. Theoretically, a wearout failure like this won't occur until after several hundred years of use, but we shorten the life span of the chips by placing them in high-temperature, high-voltage, or power-cycling environments that can cause early failure. This is the reason that the computer finally fails after so many years of faithful operation. Its parts simply wear out.

Problems between the chip leads and the support structure pins which connect the device to the rest of the computer can cause failures such as inputs or outputs shorted to ground, pins shorted to the +5-volt supply, pins shorted together, open pins, and connectors with intermittent defects. Most failures result from opens or shorts to ground. Chips fail far more often than diodes or transistors, because the chips that are the same size as single (discrete) diodes or transistors contain hundreds or thousands of tiny circuits that produce more heat and therefore more thermal wear.

HOW DISK DRIVES FAIL

Disk drives give us the ability to save and load software at almost unbelievable speeds. These "boxes" are some of the most complex collections of electronics and mechanical hardware ever constructed. Thousands of tiny magnetic signals are stored on each disk inserted into one of these drives. We expect disk drives to save all our programs and data accurately and quickly and to accurately load the information back into our IBM PC without a single lost number or letter.

And they do. Disk drives will give you months of faultless service if you do your part, operating them carefully and providing tuning and periodic cleaning.

But sometimes users operate drives while puffing on a cigarette, tapping ashes onto a tray at the side of the drive. They smile as they jam a disk into the drive and then slam the drive door closed.

Then one day, that horrible DOS ERROR message appears and the drive "gives up the ghost." Now what? What kinds of failures can occur with disk drives?

The consistently heavy use of these electromechanical machines is the reason that the most common computer system problem involves the disk drives. Not only is this circuitry active much of the time, but any machine with mechanical movement, by definition, will require periodic alignment. The moving parts in disk drives gradually drift out of very strict operating specifications.

The typical drive failure is a change in the drive rotation speed. This affects the reading and writing of information on the disks. The speed is adjusted for approximately 300 revolutions per minute–200 milliseconds per revolutions. As the speed varies from this, disk read and write errors begin to occur. Drive speed and tracking must be within prescribed limits for proper disk storage and retrieval. Drive speed and tracking must be periodically checked and adjusted because age and movement can cause the settings to drift off. You shouldn't worry about relocating your system. However, when you do, lift and set down all the computer equipment carefully.

Even when the drive sits in one place and simply operates day in and day out, the drive and tracking can shift outside proper write/read specifications. Every time you energize the PC, the system runs a diagnostic test and then commands the magnetic head in the drive to seek the home (track 0) position. When it senses home, it moves back and forth from track to track writing or reading data. The movement of the mechanical parts in the drive can affect the tracking a tiny bit with each disk access. After a

year of use, the tracking can be barely within limits. So we must periodically check and adjust the tracking and speed to keep them in "tune." The procedures for doing tracking and speed adjustments can be found in Chapter 4.

Rough handling in disk insertion and removal can cause misalignment of the read head. Misalignment is not an easy thing to fix. It usually requires special software and head alignment tools possibly including an oscilloscope.

HOW DISPLAYS FAIL

Most of us don't anticipate failure of a display monitor. Although monitors are like television sets, and we know from experience that sooner or later a TV will develop a problem and need repair.

Part of the reason displays still fail is that displays are the only new electronic device that still uses a vacuum tube. The cathode ray tube (CRT) is the screen you look at when you work with your computer–it displays video information. The CRT is probably the only modern electronic component that is guaranteed to wear out.

In Chapter 2, you learned how the letters and numbers you see on your screen are produced by electrons striking the back side of the screen. The electron streams get weaker as the CRT ages. You can correct some of the effects of age, but this requires knowledge and experience in television and monitor repair. Unless your're so trained, it's better not to open the display unit and expose yourself to those dangerous high voltages.

Here are some possible video display failures:

Short inside the CRT–can result in a "hum" noise and a bar across the screen, very poor contrast, a bright beam on the screen, or even diagonal lines on the screen.

Open or disconnect inside the CRT–no characters are displayed on the screen.

Bright "bloomy" letters; poor intensity control–caused by tube age. The center of the CRT has worn so that you can get normal brightness with the intensity turned down as far as possible, but black is really black, and grey shades are poor or not displayed.

Screen edge won't display; picture fuzzy–a deposit has formed on the inside of the screen causing reduced brightness and fuzzy display. The deposit is thicker at the outer edge of the CRT.

No picture–brightness and intensity controls have no effect.

Marginal performance–display monitor performance less than optimal. Monitors, like computers, printers, and other electronic equipment, are affected by dust and dirt. These pollutants coat the components inside the chassis and cause heat to build up. You know (now) what heat can do to your equipment.

In general, CRT failures cannot be corrected by anyone other than a trained service technician. The voltages inside the chassis of your monitor reach as high as 25,000 volts. These levels can be lethal if you make a mistake.

Unless you also repair displays, the only adjustments you should attempt are those that can be accomplished from outside the chassis. If you see holes in the back of the chassis for alignment, you'd be better off keeping out of these, too; but if you feel experimental, be sure you use a plastic alignment tool (it looks like a thin pen with screw-driver-shaped ends).

OTHER FAILURES

Liquid "Fry." This occurs when someone holds or sets a liquid on top of or too close to the computer and then accidentally spills the liquid into the top of the keyboard while the computer is running. It's a real mess to clean up, and you also get to replace many components.

Component Failure by Asphyxiation. This is caused by blocking the IBM PC vent openings

or stuffing your computer with piggyback expansion boards that produce lots of heat without considering additional cooling. It "kills" components.

The Interface that Doesn't. This can be caused by improper connection of cables. Plugging cables in one pin off blows many chips. If cable connectors are badly corroded, no signal can get through the cable.

RFI Wipeout. Ribbon cables don't have much protection from radio frequency interference or magnetic fields produced around high voltage machines or even power cords. Printers may print garbage or not at all if the ribbon cable connecting the computer to the printer runs alongside or through a loop in a power cord.

So much for "other failures." If it can be done, someone has probably done it.

REPAIR GENERATED FAILURES

Some people simply have a knack for fouling up the works every time they try to "repair" something. They should take up reading instead of repair. Overzealous or under trained repair technicians, and technicians in a hurry or not understanding the system being analyzed can introduce more trouble than they can correct.

The following paragraphs describe some possible repair-generated "failures."

Bent or Broken Pins. Watch the way you put those chips in. You can only straighten those pins so many times before they break off completely.

All too often, pins on ICs that you are inserting into the board during repair bend. One of the most frustrating things to have happen is to desolder and replace an IC only to discover much later that you bent a pin under the chip shorting out etches on the board beneath the chip. This creates symptoms that cause you to believe something else is wrong in the system. Think about what would happen if you replaced a chip causing a speaker failure and now the drive won't boot. You could spend hours searching for a failure that you created yourself! To avoid this, be alert and careful when conducting repair. Above all, don't rush fixing the problem. Take your time and make sure the job is done right. Check your work after you have replaced something.

Electrostatic Discharge

Devices can be "blown-up" by improper handling. This problem occurs when someone picks up ROM, or CPU chips without first grounding any static electricity that a person might be carrying.

Many technicians improperly ground static electricity. People and objects such as desks and benches can accumulate a substantial electrical charge. Your body can actually accumulate static charges up to 25,000 volts. It's not unusual to build up and carry charges of 500 to 1500 volts. When you touch a computer or component inside the machine, the potential on you will discharge to ground. The electrostatic discharge will find the shortest path to ground. If it's through a chip, it can damage or destroy the IC. Many types of ICs are very sensitive to static electricity. A discharge of only three volts into a chip can cause malfunction and cause wild screen displays.

Each IC in your IBM was designed to withstand a certain amount of low voltage discharge. Latch-up and the destruction of the transistors in the chip can occur when excessive ESD is passed through it. To prevent ESD problems from affecting electronic circuitry, you should discharge any potential on your body to ground before touching anything inside the machine. This can be accomplished by touching a grounded area of the system such as the power supply case. The best technique is to use a ground strap attached around your wrist and connected at the other end to system ground. This frees you to move about and touch components without fear of zapping something. Some chips, such as the bipolar TTL ICs, are more suseptible to static than others. The chips produced in metal oxide semiconductor (MOS) technology are the most susceptible. These chips include:

- The 8088 CPU
- The ROM memory chips
- The 8237 DMA Controller
- The 8253 Programmable Interval Timer
- The 8255 Programmable Peripheral Interface
- The 8259 Programmable Interrupt Controller
- The 8284 Clock Generator
- The 8288 Bus Controller

Improper Soldering/Desoldering

Another self-generated failure can be caused by improper desoldering and soldering techniques.

Caution: never attempt to desolder or solder a PC board if you don't know how!

Soldering and desoldering components is a skill that every technician should know. But, if you're a novice, for goodness sake, don't learn by practicing on a $500 system board. Practice on a board that won't cost much to replace.

Typical solder-related problems include leaving the solder pencil, or gun on the board so long it melts the etches and pads. This can introduce many more hours of repair work than you expected.

Solder "splashes" can also raise havoc on a component board. These are caused by using too much solder to fuse a connection. When you remove the pencil, a tiny ball of solder drops from the end of the soldering pencil right on top of the board, shorting out some of the circuit. Sometimes, a solder ball is so tiny, you may not even notice it falling into the circuitry.

Another potential problem with using too much solder occurs when you hold the soldering pencil to the pin being soldered and over-feed solder to the connection. The solder can flow through space in the hole around the pin and start building up on the other side of the board. It will flow through the board and then start climbing the pin. When the pin is all covered with solder, a big solder ball builds under the chip shorting the pin to other pins and board etches around it. This is a sure indication that the technician doesn't yet understand how to solder properly.

During one repair, a resistance test from one pin in the circuit to ground that should measure very low instead read nearly infinity. There was continuity across the component, but one side to the next component in the data path tested open. A visual inspection showed a bad solder connection at the component wire lead. The wire end was cleaned, retinned, and resoldered eliminating the problem and completing the repair.

In another case, many hours were spent alternately applying canned coolant and hot air to various suspected areas of the suspected circuit. The failure symptom remained. More time was expended pulling on wires and prodding joints with a plastic screwdriver blade, but to no avail. Finally, while moving one wire, the problem disappeared. The pins were resoldered correctly removing the cold solder joints and eliminating the problem.

If a component is desoldered during a test, or a new part is installed, make sure it is placed in the correct board holes and soldered carefully. Careless or improper desoldering and soldering can create troubles and multiply the difficulties converting a typical repair into a tough dog. Soldering and desoldering are covered extensively later in this chapter.

Installing the Wrong Replacement Part

I was recently told of an unusual problem that was observed by a service technician repairing a broken machine. During the visual inspection, the service technician noticed fresh solder flux around a diode and a nearby transistor on a circuit board. Someone had replaced the original diode and transistor in some vain attempt to correct the malfunction. Both the diode and transistor tested good by the technician. Then the technician noticed that the previous repair technician had installed a universal transistor in

place of the device on the schematic and in the parts list. Replacing the universal transistor with the original required part didn't change the failure symptom. Since the transistor replacement was improper, the technician inspected the replacement diode carefully and discovered that this component was also incorrect. The correct diode was installed and the system was restored to correct operation.

Max Goodstein, in an article appearing in the April 1987 issue of "Electronics Servicing & Technology," described a difficult troubleshooting incident in which a malfunction occurred about 30 minutes after circuit activation. He sprayed the components in the suspected circuit with cooling spray, hoping to identify a heat-sensitive component. Cooling one IC seemed to help, so he replaced the IC, but the problem was unchanged. Mister Goodstein tested all the components in the malfunctioning stage both in and out of circuit. No bad components were found. Then he noticed two parts in a related stage that were connected to the suspected stage. Conducting ohmmeter tests of the resistors, capacitors, and diodes in this stage revealed a shorted diode. The diode was replaced, but the symptom remained. Removing a transistor from its mounting, he measured the forward and reverse resistances with a meter. He discovered base-to-emitter leakage in the transistor. Not having a direct replacement in stock, Mister Goodstein substituted a "similar" transistor into the circuit. The malfunction was not corrected. When he installed the proper type replacement transistor, the problem went away.

Be careful to use the correct replacement parts for failed components.

Improper Cable Hookup

I don't know of any technician who at one time or another hasn't incorrectly mated cables and plugs. How many of you have hooked up a printer to the wrong adapter card and later realized the error.

Another typical error is not making a tight cable connection. The interface looked fine, but only part of the signal got through. This also happens if you haven't kept the connectors clean and corrosion blocks signal flow.

Noise Interference

Signal noise can be caused when you place cables near sources of RF interference. This noise comes from being too close to CRTs, and power and interface cables. Don't place cables near the CRT or pass them through loops of a power cable. Interface cables are insulated from most noise, but you can defeat their shielding by placing them in an EMI or RFI field. This will produce symptoms of system failure. Noise interference is a major cause of intermittent failures.

DOCUMENTING YOUR PROGRESS

Technicians are often interrupted during their analysis. Any interruption that breaks your deductive thought process and pulls you temporarily away from the problem you are troubleshooting can cause the loss of hours of crucial analysis. By the time you return to the system, you've forgotten where you were in the process. To help you quickly refocus and again concentrate on the failure, write down key points as you proceed. Then when you're interrupted, you can return later and quickly refresh where you've been and continue where you left off.

HOW TO LOCALIZE FAILURES

There are two ways to localize failures and determine which computer part is broken: the software approach, and the hardware approach. Each approach is important in its own right, so each will be covered separately in the following paragraphs.

Software Approach

The software approach is a troubleshooting method used widely by most IBM PC repair

technicians. As long as the disk drive will boot up properly, we can often find the failure using diagnostic software. As you know, your PC has a built-in diagnostic software program that checks out the machine each time you apply power. This program is well written and does much to ease your mind that all is well inside. More on this later.

Diagnostic Software

Watching strange things happen to a computer system can be frustrating. Often you can't be sure if you caused those weird characters on the screen or if your IBM PC is truly sick. You'd rather not start taking the system apart for failure analysis if the machine isn't really broken.

There is a way to gain confidence that the system is healthy and that the errors are probably in the software program you're trying to run. If the error is repeatable and the system drive still boots up, you can insert a diagnostic disk into your PC system and run a series of programs that test the condition of the computer. These self-test routines can give you a 95 percent or greater confidence indicator that your PC is working properly and that you need to check your software.

Diagnostic programs can also show possible faults before they become hardware problems. For example, some diagnostic software tells if the disk speed is too fast, too slow, or within a speed range where reading and writing data can occur without errors. These diagnostics measure the mechanical operation of your disk drives and are helpful in periodic preventive maintenance.

The success of self-test packages is measured by the level of confidence one can have that the component identified as bad by the software is indeed faulty. Some diagnostics are advertised as only 60 percent accurate; other companies say that their software test packages have an 85 percent confidence factor.

Most minicomputer diagnostics only identify faults to the board or module level. That's because customers in the large companies that own most minicomputers usually depend on the computer manufacturer's field service representatives for repair support. Here, the diagnostic is used as an improved user interface. The user can relay to the computer service center what the diagnostic tests have determined giving the field service technicians a quicker troubleshooting and repair visit. This is exactly the situation with your IBM PC. If it fails during diagnostic testing, a number is printed on the screen. This number is a key number to help you identify the bad part. (As Chapter 4 will describe, you can use this number to your advantage also).

Fortunately, most of the IBM PC microcomputer diagnostics can call-out faults to the chip level (especially faults in memory).

About 30 percent of all PC failures can be detected by diagnostic programs. Diagnostic programs can be bought from many of the popular IBM computer stores. The IBM advanced diagnostics is the best documented of them all. It comes complete with a hardware maintenance and service manual that contains a complete list of all the power-on self test error codes that could be displayed when energizing your IBM. There is a similar list in Chapter 4 of this book.

The IBM advanced diagnostic package also contains a switch configuration chart and a problem isolation chart to help you quickly get to the failing module. The diagnostics don't contain maximum stress tests for the floppy disk drive, the memory, or any hard disk drives. Therefore, if any marginal systems problems exist, this diagnostics will probably not catch it. The diagnostics software that comes with each IBM system is much like the Advanced version, but doesn't contain the option to format the hard disk drive, or the wrap plugs to test the asynchronous and printer adapter. In addition, little documentation is provided with the diagnostics that comes with the basic system.

Besides the diagnostic software provided with the machine, several companies provide diagnostic programs for IBM PC. These programs test main memory, system read-only memory (ROM), the CPU, the monitor, the keyboard, the disk drive speed, and many peripherals.

The most common diagnostic programs check the system random-access memory (RAM) and some of the input/output. Some routines check the operation of the CPU itself, but these usually locate only minor errors. It's difficult for a CPU like the 8088 to run a test on itself. Most diagnostics assume that the CPU is working properly.

Testing a microcomputer must begin with thorough analysis and test development related to each component in the design. The logic of this approach is that verification of satisfactory performance of each component must be assured before a test is conducted on the whole system.

There are three widely used methods for testing a microprocessor like the 8088 CPU. These are:

1. Actual use
2. Stored response from a known good system board
3. Algorithmic pattern generation

In actual use testing, a very limited subset of the CPU's capability is tested under generally ideal conditions. This test technique is satisfactory for noncritical applications, but is not suitable for critical applications such as braking systems and medical monitoring and controlling systems.

Storing a set of response vectors derived from testing a known good board also has some limitations. How was the reference system board verified good? Another limitation is the large amount of memory required to store simulation and response patterns.

Generating an algorithmic pattern gets around much of the large storage requirement. A test vector compaction technique such as signature analysis further reduces the memory requirements of the tester.

Without automatic test equipment, the CPU is usually tested by executing a software program that exercises the nonmemory parts of the chip such as the data path, the peripheral I/O logic, and the sequencer. During the execution of the program, the test results are compared to a pattern of expected results after every stimulus is applied to these functional blocks. Typically, an external tester monitors the output pins and compares the readings measured with stored expected values.

Testing the I/O Logic

The I/O logic in the IBM PC consists of several latches and transceivers that enable 8-bit data and 20-bit addresses to move about on the system board. A test program can be written that exercises individual ports in a predetermined manner. By monitoring the port, and knowing what information should be present on the pins, the output signals can be validated.

Testing the Interrupt Logic

The interrupt flag and interrupt enable latches are controllable and observable. Therefore, functional test patterns can be generated and applied to this circuitry via a test program written specifically for this purpose. Verification of the proper interrupt sequence can be easily made, but to check the priority logic, an external tester will likely be required so the output signals can be indirectly verified by checking whether or not a proper interrupt service routine was executed.

Bus Testing

The bus connects many devices on common interconnect lines. The first step in functionally testing a system board is to make sure the bus structure is free of defects.

Two types of problems are associated with a bus. First, problems occur with devices exhibiting leaky outputs associated with weak internal diodes. Second, input problems occur in connected devices that have internal shorts that overdraw current from the bus.

The IBM PC employs many buses for data communication, including the control bus, the operand and result buses in the data path, the memory address and data buses and several buffered buses on the system board and each of the peripheral boards. By controlling and observing a bus, all the logic connected to it can be easily accessed, improving their testability.

No special test points are needed to access the devices connected to a bus. In addition, all registers connected to a bus can be accessed by accessing the bus. One good access place on the system board is the expansion slot backplane. By inserting an extender card into one of these slots, many bus signals are readily accessible. By accessing one bus line at a time and observing the results, a bad bus line can be confirmed.

One technique for isolating bus output problems is to get the bus into a tristate condition with all bus-connected devices disabled. Then use a tester to pull each bus line both high and low. If the bus passes this test, the bus is good. If the bus does not pass this test, check each device connected to the faulty bus. Run that device to a pin state opposite the state in which the bus line failed. By measuring the difference in the amount of current required to bring the bus line back to the opposite pin state, a faulty device can be detected.

If the bus problem is an input problem, inject current into each device connected to the bus and detect the current flow on the bus. In this manner, input faults on bus-connected devices can be detected.

Memory Tests

Some memory diagnostics test to see if the computer is properly setting and clearing individual bits in memory and also if store or write operations are affecting more than one memory address location at one time. Other diagnostics test the permanent memory (ROM) by reading every location and then computing a final signature such as a checksum or a cyclic redundancy check code.

Both the read only and read and write memories on the IBM PC system board are tested during the boot-up process. Each type of memory test is a part of the ROM BIOS power-up program.

ROM Diagnostics–In the IBM PC, each ROM is tested using the checksum technique. Initially, the 8K ROM containing the BIOS is read and its contents summed. The final summation is compared to a stored value. If an error occurs, the system halts with an error message displayed on the screen.

The routine begins by disabling the NMI interrupts, initializing the DMA page register, disabling black and white and color video, setting the 8255 PPI A, B, and C ports for A and C as input and B as output, writing the 8255 command code register, disabling the parity check, setting up the data segment register in the 8088 to point to the ROM address, setting up the ROM starting address at E0000H, setting up the return address, and jumping into a ROS CHECKSUM routine.

Later in the start-up routine, a check is made for an optional ROM at address C8000H through F4000H. If one is found, a checksum routine is conducted on this ROM. Otherwise the program jumps to the next 8K ROM module and a checksum is performed on this chip. After all the remaining ROM modules are tested, the power-up routine continues (checks to see if a disk drive is attached).

RAM Diagnostics–Testing RAM memory is more involved than testing ROM because RAM requires the write operation besides the memory cell read.

The main memory tests assume the CPU is fine and go on to do some fancy tests on the RAM. This form of testing finds out if test data can be correctly loaded into one and only one location in memory. If a "storage error" occurs–that is, the test data stored is not the same as the test data–a message is printed on the screen. If the correct data gets stored but into several different memory locations at the same time, an "addressing error" has occurred and this too is noted on the screen.

There are many algorithms (routines) for testing memories. Typical RAM tests are machine language programs that carry out RAM write-read algorithms. RAM is tested by a conventional memory testing algorithm carried out as a software program that writes a pattern of data into memory, then reads the value out and compares it with an expected value. Because the test results are compared after each write-read

operation, a RAM test takes time to complete. In addition, by carrying out a machine instruction write-read algorithm means that the RAM cannot be exercised at its full speed capability. Therefore, the test, while verifying the ability to write and read properly, does not guarantee that the same performance is assured at full operating speed. The following is a list of the most common memory tests:

Common Memory Tests
Simple Store and Read
Sequential Numbers Test
Rotating Bit Test
Walking Bit Test
Dual Address Test
Butterfield Test
Sum Test

A "simple store and read test" preserves a known value in every location in a selected block of memory. Then it reads the contents of each location to ensure that the value was correctly stored. It is a quick and easy rough test.

A "sequential numbers test" involves loading all the binary number combinations for an 8-bit word sequentially into a block of 256 memory locations. Then it starts at the first address location and reads out the data word stored, comparing it to the value that should be there. If the data is correct, the routine displays the words "all O.K." and the test moves on to the second location. If an error is found, the program displays an "error" symbol on the screen and the test starts over at the next (third) address location. The test repeats until you reset your system.

A better memory test, the "rotating bit test" checks each address location to see if a binary bit stored in any one of the eight positions in a binary 8-bit data word will falsely set another bit in the same word. This test starts by loading the binary number 0000 0001 in the lowest RAM address. The contents of this address are then read back out and verified. If the 0000 0001 was correctly stored, the bit is shifted left one place to 0000 0010 and the test is repeated. After the set bit (the "1") is shifted through all the binary combinations, stored in that same address location, read out, and verified, the entire test starts over at the next memory address location.

The "walking bit test" improves on the rotating bit test slightly. All 8 bits in a starting location are set to 0, or "cleared." Then the first bit is set to "1" (0000 0001) as in the rotating bit test. The program tests all seven other bits to see if they have changed from 0 to 1. Then the second bit position is set to 1 and all other positions to 0 (0000 0010). Again all seven other bit positions are tested. This process walks through each bit in that memory location setting each bit to 1 and testing all seven other positions.

Then the values are all reversed; all the cleared bits are set to 1 and the set bits are cleared to 0, and the entire process begins once more, but now as a rotating zero test.

This test is quite time consuming. Apparently, it can take over 13 hours to check a 16K-byte area of RAM. And it can take over 52 hours to test 32K bytes of memory! You can just imagine how long it would take to test a fully packed IBM PC.

A "dual-address test" provides a more thorough addressing check. Starting with the lowest memory address in a selected block of memory, the program stores all zeroes into the area (clears it to zero). It then stores all ones (1111 1111) into the first location and checks all other locations to see if any other memory address falsely received any ones. If all other locations are still "zero-loaded," the test location is cleared (written into with all zeroes) and the test shifts to the next higher address, storing all ones in this location and then testing all other memory locations. This test repeats until the program reaches the end of the selected memory area.

Jim Butterfield wrote a program that is a variation of the dual-address test and is in the public domain. In the "Butterfield test" program, all ones are stored in every location of the selected memory area. Then all zeroes are stored in every third address location starting with the first address. The algorithm then checks the contents of every memory address to make sure the values have been stored correctly.

Next, the program shifts the position of the "all zeroes" word twice using the second and

then third locations in the memory as starting points. After the three-pass test using 0s in a memory field of all 1s, the bits are reversed and all 1s are stored in every third location of an all 0s memory field.

If an error is found, the program stops and the address of the error is displayed. If no error is detected, the program ends and the top address plus one is displayed on the monitor.

The "sum test" is probably the most sophisticated memory diagnostic test. It generates a unique data word for storing in each location of memory to be checked. The data word is the sum of the two bytes that comprise that memory address (recall that it takes 16 bits to address 64K bytes of memory; 16 bits is two 8-bit bytes). Since each succeeding address is one location higher, the value stored increases and each value is unique to an address. A variation on this scheme can be used with the 20-bit address word in the PC.

The algorithm then checks for correct value storage. If an error is found, the program displays the error and its location on the screen.

This diagnostic test is also time consuming. It's a good idea to run these types of dual-address tests on small blocks of memory rather than testing all the RAM. It has been determined that the testing time quadruples for each doubling of the amount of memory tested.

The RAM memory test in the IBM PC is a write/read/verify operation in which the patterns FF, 55, AA, 01, and 00 are written into the first bank of memory. After each write, a read is executed and the value fetched is compared to verify proper storage and retrieval.

Self-Diagnosis

There is a trend toward building diagnostic capability into peripheral equipment like printers and plotters. A strong incentive exists to place diagnostics in CRT displays, disk drives, and the personal computers, because so many of these devices are being sold.

Disk drives and printers function both electronically and mechanically. The electronic controller portion of these machines can contain their own diagnostics, and many controllers now do some form of self-diagnosis each time the system is powered up. These tests check for faults in the electronics.

Mechanical components are inherently less reliable than electronics, so peripherals containing mechanical parts need diagnostics that regularly check their internal operation. Most of the conditions monitored are operator related; for example, "paper out" or "ribbon out." Disk drive diagnostics measure mechanical parameters like speed and head alignment. We cover disk speed adjustments and head alignment in Chapter 4.

All the "canned" diagnostic packages use some version of the seven test algorithms described previously. Each diagnostic program is a valuable addition to your "troubleshooting toolbox," but no software diagnostic can help if your system won't boot or display. The message is: "There are many ways to skin a computer cat. Know them all."

Hardware Approach

In the hardware approach, troubleshooting tools are used to measure voltage (logic) levels in the circuitry of the IBM PC. These tools include the logic probe, the logic pulser, the current probe, the oscilloscope, the multimeter, the logic analyzer, and the signature analyzer. This approach requires a knowledge of electronics and test equipment.

Usually when a chip comes to the end of its useful life, a catastrophic failure occurs–it cooks itself internally. While your eye can't always see the chip defect, you can find the problem without much effort. (But, don't think that every time your IBM PC quits working, you've just had a catastrophic failure.) For those problems that are not easy to identify, let's refer again to our guidelines for success.

1. *Don't Panic.* You now have a manual that will help.
2. *Observe.* What are the symptoms? What conditions existed at the time of failure? What actions were in progress? What program was running? What was on

the display screen? Was there an error message?

3. *Sense.* Is there any odor present from overheated components? Does any part of the system feel overly hot? Do any of the components look charred or broken?
4. *Retry.* If the display is dark, check the brightness control, the power plug, and the power cord. Is the plug snug in the back of the computer? Is the other end of the power cord plugged into a wall socket? Is the wall socket working? If any of these isn't right, correct the problem and try again.

   If the problem involves an external display, the printer, or other I/O peripheral equipment connected to the IBM PC by cable, make sure the power to the system is off, disconnect the power plug from the computer, and then reseat all the connector cables associated with the failure. Cables have a habit of working loose if they aren't clamped down. Once you've checked the cable connections, reconnect the power plug, power up, and retry.

   If a disk didn't boot, try booting the disk in the other drive or try booting another copy of the program disk. You could also try booting the disk in another IBM PC computer. If you always use a copy of the program disk, any failure of a disk drive won't cause as much frustration if it destroys data on the backup disk as it would if the disk were the program master. If data is altered by a malfunctioning drive, the disk can be recopied again from the program master once the drive problem is resolved.

   If it still won't work, disconnect all the external equipment connected to the computer, and try to operate the system alone. Sometimes, the failure in a peripheral device appears in another functional part of the computer. If the computer works by itself, the problem is probably in the external device or in the connecting interface.
5. *Write.* Document all that you see and sense. Write down all the conditions that you observed at the time of failure, or when you verified a reported failure. Write down what conditions exist now that failure has occurred.

   What is the PC doing?

   What is it not doing?

   What is being displayed?

   Is there an error message?

   What is still operating?

   Is power still indicated on each part of the system?
6. *Assume one problem.* In digital circuitry, the likelihood of multiple simultaneous failures is low. Usually, a single chip malfunctions, causing one or more symptoms; however, if you've shorted something in the circuitry, all bets are off.
7. *Use correct service data.* You'd be amazed at how many small service centers are trying to run repair operations with little or no technical information on the equipment they claim to support. In one case, a major service center covering an entire state was conducting repair activities on a myriad of personal computers with a 20-page "technical manual" and one of my troubleshooting and repair guides. The manufacturer's "technical manual" was so poor it was essentially useless. Because the micro maintenance series of trouble-repair guides are high level descriptions of personal computer equipment, the service technicians had no source for accurate measurements, waveforms, DC voltages, and in-depth technical theory of operations for the machines.

   Don't attempt repair without proper service data. Make sure you have the SAMS COMPUTERFACTS, this manual, and anything the manufacturer provides at your bench. The use of correct and complete service information can prevent moderately difficult repair jobs from becoming tough dogs. If you value your time, prepare before you repair.

8. *Use the right test equipment.* Just like the proper technical documentation, the right test equipment can change difficult repair jobs (dogs) into routine activities. Unfortunately, all too many small service centers are trying to conduct a repair business with old, substandard, uncalibrated equipment. How many of you have worked (or are now working) in a repair center that has little more than a VTVM and some hand tools. Going after a failure in electronic circuitry inadequately prepared is like tracking a rabbit through your carrot garden with your eyes blindfolded. Your actions make little difference and all the while that rabbit eats more and more of your garden away.

   The lack of essential test equipment such as logic probes, pulsers, and a good triggered scope makes a world of difference in computer repair. If good troubleshooting equipment is called for, nothing less will do. Visual examination of the condition of the machine and its operation can have great value (when they're successful). However, this won't locate serious defects. Therefore, limit the time for simple tests to a few minutes before you change to using a DMM, scope, or generator. Be careful you don't apply inappropriate troubleshooting resources to the problem.

9. *Diagnose to section.* If the system worked when the peripherals were disconnected, turn the power off and reconnect one of the peripherals. Power up and test. If the unit still works, turn the power off, and reconnect another peripheral. Power up and test. Follow this procedure until the unit fails. The built-in diagnostic tests are a big help here. Once failure occurs, you know what device and what interface section has the problem.

   If you disconnect all the peripherals and test the computer alone, and it still won't work, try to determine what section or division of the machine failed. Describe the failure in simple terms–drive B won't read a disk.

10. *Consult the symptom index.* Chapter 5 includes an index of the most common troubles with the IBM PC. It includes a section on system error displays. If any error codes are displayed, these self-diagnostic results can guide you to the correct area of the problem. If the symptoms that you see match a problem described in the "Troubleshooting Index," turn to the referenced page and follow the instructions under "Troubleshooting Procedure."

Caution: Any time you open the computer, ensure the power is off, and touch a metal lamp, or other grounded object, to remove any stray static electricity.

11. *Localize to a stage.* Turn off the power to the computer, and disconnect the power plug. Disassemble the computer as shown in Appendix D. Follow the troubleshooting steps and procedures in Chapter 5 to localize the failed stage.

12. *Isolate to failed part.* Closely following the procedures in Chapter 5 should guide you to the failed part.

Many things get in the way of proper system operation. Chips have a tendency to work themselves out of their sockets under normal operation. A loose RAM chip could be your whole problem. "Loose chips sink MIPS" (MIPS stands for millions of instructions per second–a measure of computer capability).

Replacement of chips that are in sockets may look easy, but there are some pitfalls you should be aware of. Those fragile pins on your chips bend easily, and it doesn't take very many straightening actions to break a pin completely off.

Removing and reinstalling chips that are soldered into the motherboard are actions that require more than a passing knowledge of soldering techniques. Only attempt this part of the test-repair procedure if you have experience soldering and desoldering multilevel printed-circuit boards. If you don't have the experience, get it–your job requires it.

Sometimes a problem is caused by noise. Not audible noise, but electrical noise, the kind that produces "static" on your radio. This noise also affects computers. Noise in the computer system can cause data to be lost or wrong data to be stored or displayed.

> Note: To avoid noise problems, keep cables clear and away from power cords, especially coiled power cords.

And it's appropriate to add, don't try out your new drill set next to your computer while computing the effects of your recent pay raise. Your calculations might prove unbelievable.

Intermittent Failures

Sooner or later you're going to be confronted with those once-in-a-while failures called "intermittents." These can be really frustrating. It's no coincidence that most "tough dog" problems are associated with intermittent symptoms. Erratic operation multiplies the difficulties.

Unlike a hard (constant) failure, an intermittent problem shows up randomly, or only at certain times (usually when you expect it least). Intermittent failures are difficult to handle using standard troubleshooting methods to obtain dependable, repeatable and easily interpreted results.

Since intermittent failures can be caused by shock, vibration, or temperature change, these conditions can be used to find and correct them. Here are some helpful hints regarding intermittent failures:

> **Caution: The following steps are conducted with the computer open and operating.**

Be careful not to short out any connectors or pin leads. Use only a nonmetallic or wood object to probe components inside an energized IBM computer.

a. Check, clean, and reseat all connector boards and cable plugs.

b. Tap gently at specific components on the suspected board using a nonmetallic rod or plastic screwdriver.

c. Heat the suspected area with an infrared lamp or hair dryer. Don't overheat it.

d. Spray coolant on a suspected component when investigating an intermittent failure. Several companies sell pressurized cans of coolant spray that have long plastic extender nozzles for pinpoint application By cooling the device with the computer energized, and operating the system, you can identify heat sensitive chips on the verge of total failure. A system with a heat-sensitive chip will begin working for a few moments after cooling until the chip heats back up and starts causing problems again.

e. After you've found the area where the problem is located, make sure the power is off, and use a strong light and a magnifying glass to look for small cracks in the wiring or solder connections.

If the problem is a marginal chip, replace the pesky rascal. Be sure your replacement chip is of the same logic family as the original (that is, replace 74LS74 with another 74LS74).

Good cleaning, pin and board reseating, and inside-the-case temperature control will prevent the occurrence of most random failures. Board reseating is not a problem on the PC since the boards can be secured down with screws.

The final method for fault isolation to a component is signal tracing. This technique will be covered shortly.

13. *Repair.* A disassembly and reassembly guide is located in Appendixes D and E.

It takes a little practice before you can remove a socketed chip without it jumping out, flipping in midair and sticking you right in the thumb or index finger with that double row of tooth-like pins. Fortunately, there are several devices that make the job much easier. These are the tiny screwdriver, or "tweaker," and the

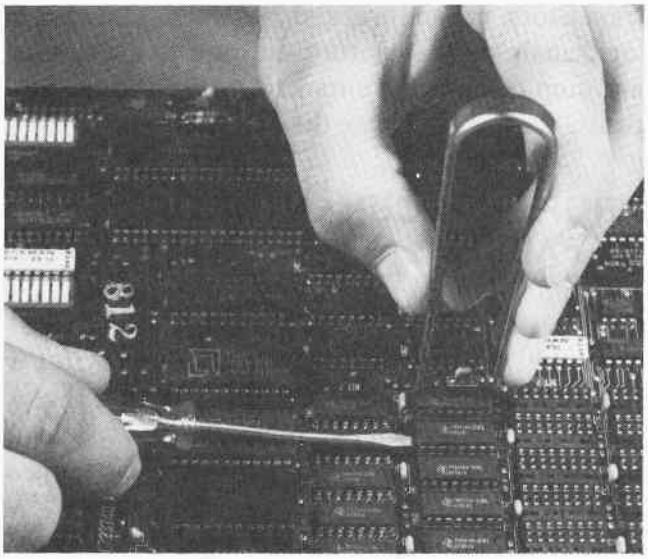


Fig. 3-4. ICs can be removed with a chip extractor or by gently prying up with a tiny screwdriver (tweaker).

IC extractor tool. Fig. 3-4 shows how each tool can aid in removing stubborn chips from their sockets.

Getting the chip out is only part of the repair challenge. Now you have to put the new chip in the socket. Here's how to do it:

a. Line up the pin-1 end (with the notch or dot) with pin 1 on the socket. (Notice how all the other chips around this socket are mounted.)

b. Place the chip over the socket, lining up one row of pins with its socket holes, as shown in Fig. 3-5.

c. With the chip at a slight angle, press down gently, causing the row of pins in contact with the socket to bend slightly, which lets the other row of pins slip easily into their sockets, as shown in Fig. 3-6.

d. Press the top of the chip down firmly to seat the chip completely into the socket. Be careful not to flex the board too much. If necessary, support the motherboard with the fingers of your other hand as you press the chip into place.

Now, that wasn't too bad was it? Well, it is pretty easy to make mistakes in chip replacement.

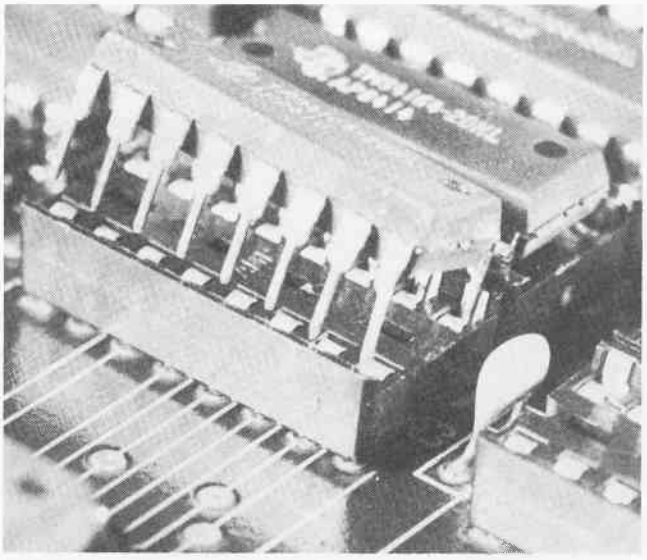


Fig. 3-5. Place the chip over the socket as shown.

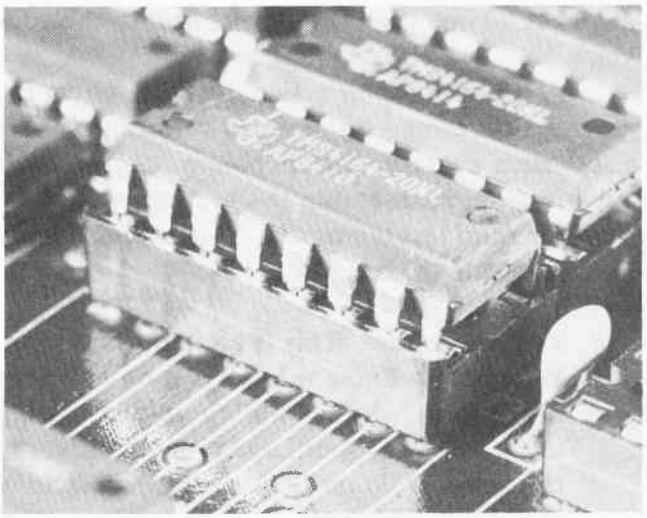


Fig. 3-6. Once each row of pins has been started into the socket, press down gently to complete the chip insertion.

- Make sure you don't put the chip in backwards. The notch or dot marking the pin-1 end of the chip is intended to help you correctly line up pin 1 on the chip with pin 1 on the socket.
- Don't offset the chip over the socket by one pin, as shown in Fig. 3-7.
- Don't force the chip down so one of the pins hangs out over the socket or is bent up under the chip.

If a chip to be replaced is soldered into the motherboard, always replace the chip with

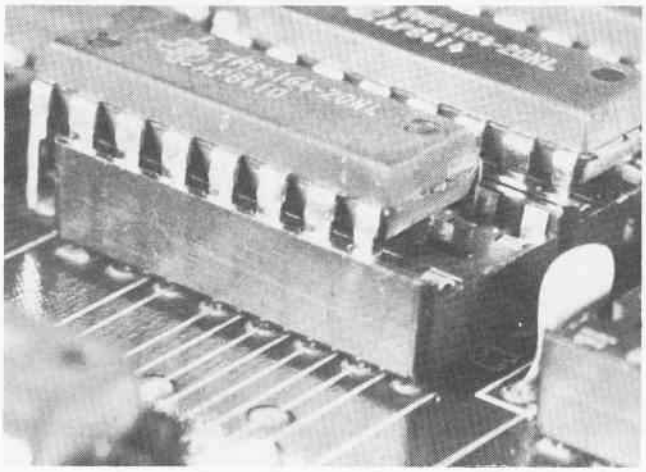


Fig. 3-7. Be careful not to offset the chip by one pin.

a socket. Then plug the new chip into the socket.

After each repair action, test the system for correct operation. Sometimes substitution of a good chip corrects the problem. After each substitution, reassemble the system enough to power up and test the repair. This process is very likely to locate the trouble.

14. *Test and verify.* This is an important step. We need to know that all is now well with the system. After booting up and testing using a copy of your DOS program disk, run the same program that was in the machine at the time of failure.

> Note: It's a good idea to log the repair action in a record book to develop a history of the maintenance conducted on the machine. Record sheets are included at the back of this book.

If these troubleshooting steps still don't help you find the failed component, hook up some test equipment, open the schematics, and start localizing the problem to a stage and then isolate the problem to a failed component. Try signal tracing with a logic probe and a logic clip. Use an oscilloscope and a digital voltmeter (DVM) to test the discrete components such as transistors, capacitors, and resistors. Connect a logic analyzer or signature analyzer to the system and step through the circuitry. Make voltage and resistance tests to locate the bad part.

PROPER DOCUMENTATION OF FAULTS

Of great value to the technician is a historical record of failures and repair actions. After a period of successful repairs, you will find that a pattern develops that can be referenced to speed up future repairs of systems with similar failure symptoms.

The following checklist is a guide to be used during and after repair. Each step is expanded here for clarity.

VALIDATING THE PROBLEM

1. What is affected? Find out if the problem is "catastrophic" and affects the operation of everything. If it affects only a part of the system (such as a disk drive), you may be able to swap drives and work on the bad drive away from the system so computer work can continue during repair.
2. Is the problem in software? Be certain it isn't. Try to run a program that you know is good.
3. Was the problem caused by operator error? Try a different operation that uses the same hardware or function.
4. Is it an intermittent failure? If your problem is intermittent, it could take several days for the problem to reoccur, and be found and fixed. You may just want to live with the problem until the intermittent becomes permanent (a hard failure). At least then you will have something concrete to troubleshoot.
5. Completely document the problem. Write down a complete description of the problem.
   - What was the system doing at the time of failure?

- What were you doing at the time of failure?
- What is the system doing now?
- What isn't it doing now?
- Is there an error code?

Maintain a written description for future reference and to build a fault history on your equipment.

6. Log the serial numbers of all the peripherals and the computer system.
7. Make a detailed listing of what was repaired or replaced.
8. Test run the system before placing it back in service.

RECOMMENDED SAFETY PRECAUTIONS

As with all devices that use or operate on electrical power, you must observe certain precautions to prevent damage to yourself or to the IBM PC system.

- Keep out of the display chassis.
- Be very careful when troubleshooting the power supply. If you're not experienced in high voltage circuitry, stay out.
- Turn the power off, ground yourself against static electricity, and pull the plug when doing anything inside the chassis except energized system troubleshooting.
- Handle diskettes carefully.
- Don't cycle the power quickly.
- Use a power strip to apply power to all components except for hard disk drives.
- Keep liquids away from the computer.
- Handle components with care.

Observing these precautions can save you time, money, and frustration. For your benefit, each point is expanded in the following paragraphs.

Keep out of the display monitor chassis. The voltages inside a monitor or television are dangerous, and only trained technicians should ever troubleshoot and repair a display unit. Voltages as high as 25,000 volts hide in there, so, unless you're experienced in high-voltage circuits, stay out!

Be very careful when troubleshooting the power supply. These circuits convert lethal 115-volt line power to the 5 and 12 volts used by the motherboard. IBM uses some special screws to secure the metal shield over the power supply. If you're not experienced in power supply circuits, stay out!

Always turn the power off, touch a grounded metal object like a desk lamp, and then pull out the power cord before touching anything inside. Before connecting test equipment to a circuit under test, turn off the power to the system. Failures are caused by people who don't follow this rule.

Handle diskettes (often called disks) carefully. Don't write on a label once it's attached to the disk jacket. Don't lay disks on a dusty, dirty surface. Keep cigarette ash away from disks and the computer system. Don't touch the disk surface. Don't try to see how flexible a floppy disk is. Don't set your disks on, or in front of, a TV or color monitor. Keep them out of strong magnetic fields.

Don't cycle the power on and off quickly. Wait 7 to 10 seconds to let the capacitors in the power supply discharge fully and the circuits to return to a stable (quiescent) condition.

Use a power strip. This saves wear and tear on the PC's on/off switch. Most power strips also have a built-in overload protection for voltage spikes. Voltage spikes can harm your computer system. (Don't connect a hard disk drive to the same power strip if it must be energized and up to speed before the computer gets turned on.)

Keep liquids away from the keyboard. I once had the opportunity to help a friend who's son had spilled a soda on the keyboard. It's amazing how sticky soda becomes after frying components all over the inside of the keyboard.

Handle components with care. Don't let chips lie around. The pins will get bent. If you lay a chip on the bench and forget about it, you

may later lean over and stick all the pins on the package in your arm. Not dangerous, just hurts. What is dangerous (to the chip that is) are indiscriminate handling of MOS or CMOS parts. Watch out for static electricity–chips may need "special handling."

SPECIAL HANDLING

Some logic devices require extra care when you touch or handle them. You have no problem removing or inserting TTL (74xx series) chips into circuit boards. But the metal oxide semiconductor (MOS) chip family (MOS, CMOS, NMOS, and so forth) need extra care because they're more susceptible to static electricity than TTL. There are some 74HCxx chips being sold today that are CMOS, but these aren't currently used in the IBM PC computer.

Don't be afraid to touch the chips in your computer. Most guides for handling MOS chips lean far toward the super-safe zone and sometimes cause more problems than they prevent. However, these chips can be damaged by the static charge you can build up by scuffing your feet across a carpet; so be sure to ground yourself by touching a metal lamp or grounded object before you reach for a chip inside the IBM PC chassis. In addition, conductive foam provides static charge protection during storing or transporting of MOS-type chips.

Additional precautions should be observed when you use test equipment with your IBM computer. Turn off the system power before connecting or disconnecting test equipment such as oscilloscopes or logic analyzers.

ADVANCED TROUBLE-SHOOTING TECHNIQUES

In this section you will learn advanced troubleshooting techniques and become more familiar with the repair technician's "tools of the trade."

Tools of the Trade

When the problem can't be solved using flowcharts and pictures, repair technicians reach for help–they reach for their "tools." These tools are not only the tiny screwdrivers (tweakers), the diagonal cutters (dykes), and the soldering pencil. They also include some electronic test equipment–the various measurement meters (VOM, DVM, DMM), logic probes, logic pulsers, current tracers, clips, oscilloscopes, and logic signature analyzers.

Meters

Electronic measurement equipment has improved a great deal over the years, markedly improving your ability to test and locate circuit troubles. Twenty years ago, a meter called a VOM (volt-ohm-millimeter) was used to measure the three parameters of an electric circuit: voltage, resistance, and current. Then came the VTVM (vacuum-tube voltmeter). It wasn't long before electric circuits made room for electronic circuits, where digital was replaced and new meters appeared for troubleshooting using some of this new capability in their design. The DVM (digital voltmeter) and DMM (digital multimeter) quickly became the preferred measurement devices for digital technicians because they offered capabilities better suited for electronic circuit testing, including increased accuracy. These meters have characteristically high input impedances (resistances) so don't load down or draw down a digital circuit where the voltages and currents are far lower than those found in analog circuits.

Two changes affected the types of tools used in troubleshooting and repair. First, vacuum tubes were replaced by solid-state devices such as transistors and the integrated circuit (IC), or chip. Second, circuits themselves became smaller with more components packed compactly into less board area. One need only compare the early radios and televisions (standing 4 feet tall and weighing 40 pounds) with the wrist radios and now the wrist televisions of today to recognize that electronic circuits are

smaller, more complex, and more difficult for test-probe access.

Electronic advances always lead to electronic opportunities, and clever test equipment designers soon came up with devices that enabled digital circuit testing without fear of inaccurate readings caused by circuit overload, or circuit failure caused by bulky test probes shorting two pins or wires on a packed printed-circuit board.

Logic Clip

One digital circuit testing device is the logic clip shown in Fig. 3-8.

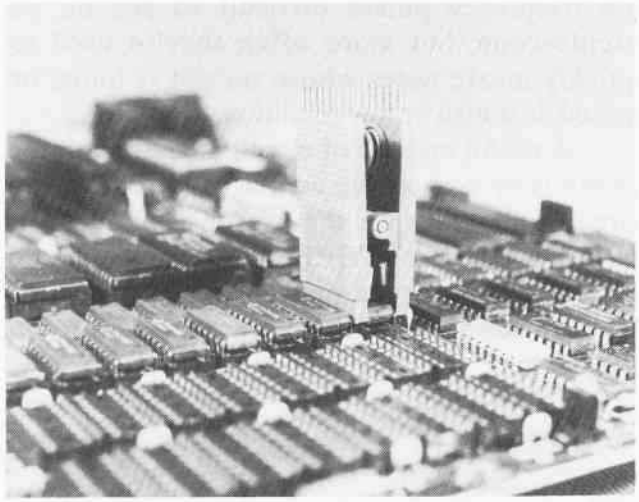


Fig. 3-8. A popular type of logic clip. *(Courtesy Pomona Electronics Division of International Telephone and Telegraph Corporation)*

> **Caution: When using a logic clip, turn power to the circuit off, attach the clip, and then turn power on. (This helps prevent accidentally shorting out the chip.)**

This handy tool fits over an IC and has exposed pins at the top. Measuring or monitoring probes or tiny clips can be attached to the pins to determine the logic level on any pin of the device under test.

Another type of logic clip has a built-in monitoring capability (Fig. 3 9). Instead of exposed pins, the top of the clip is lined with two rows of light emitting diodes (LEDs) which continuously display the logic condition of each pin on the chip. The LEDs are turned on (showing a Logic 1) by power from the circuit under test. All the pins are electrically buffered so the clip doesn't load down the circuit being tested.

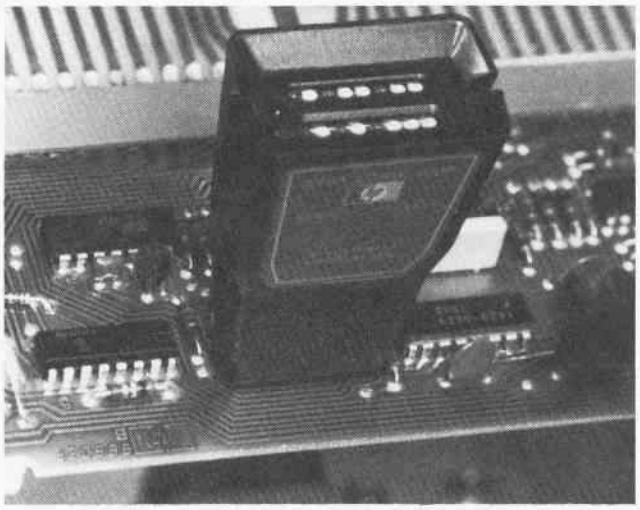


Fig. 3-9. A logic clip that gives visual indication of the logic condition at each pin. *(Courtesy Hewlett-Packard)*

Logic clips can be obtained in several varieties–to work with almost all logic families, including TTL and CMOS–and in voltages up to 30 volts DC.

To use the clip, squeeze the top (LED) end to spread the pin contacts, and slip the clip over the top of the chip to be tested. When power is applied to the circuit, the LEDs will show the logic level at each pin on the chip.

Logic clips can be used on ICs with up to 16 pins, or 80 percent of the ICs on your IBM PC system board.

Logic Probe

When you want to really "get into" your circuit, you can use a logic probe. A blown chip can't be repaired, but the logic probe can tell you which chip has failed so you can replace it.

The logic probe shown in Fig. 3-10 is the most widely used tool for this type of analysis. It can't do many of the things complex test equipment such as logic analyzers can do; how-

ever, the high frequency of chip failures in electronic circuits, the simplicity of the probe, and the ability to rapidly troubleshoot in an energized circuit make this tool ideal for 90 percent of your fault isolation needs. The SAMS COMPUTERFACTS on the IBM PC contains extensive logic probe values for use in analyzing section and stage circuitry.

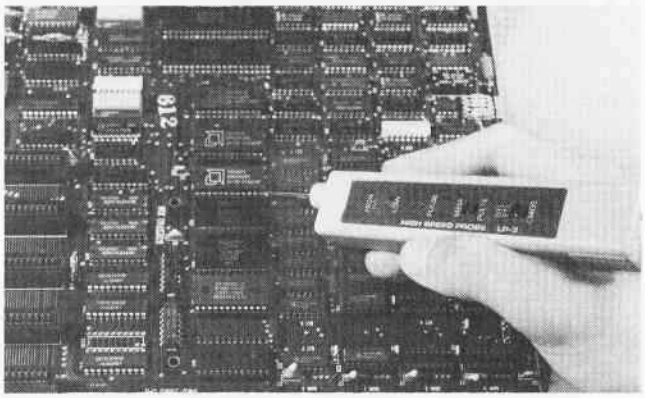


Fig. 3-10. The logic probe is the most widely used tool for circuit board analysis.

When the tiny tip of the probe is placed against a pin on a suspected bad chip, a test point, or even a trace on a circuit board, an indicator light near the tip of the probe tells you the logic state (level) at that point. The metal tip on most logic probes sold today is protected against damage from accidentally touching a source of higher voltage (up to 120 volts AC for 30 seconds) than that of logic gates (+5 volts).

Some probes have two lights built in near their tips–one for logic high and the other for logic low. The better probes can also tell you whether the test point has a pulsing signal present. They can also store a short pulse burst to tell you if a glitch or spike has occurred at that point. If you're planning to buy a logic probe, be sure it will work with the logic families you plan to analyze.

The ability to touch a point with the probe tip and directly determine the condition at the point for diagnostic analysis, and the ability to store pulses make this device easy to use and universally accepted as the proper diagnostic tool for all but the most complex digital troubleshooting. Other tools force you to attach the measurement probe and then look away at some display to read the condition. The logic probe displays the condition near the tip of the probe itself.

The logic probe in Fig. 3-10 provides four indications:

Lamp off for logic low (logic 0)
Lamp on bright for high (logic 1)
Lamp dim for floating or tristate
Lamp flashing for pulsing signals

Power for the probe comes from a clip attached to a voltage point on the circuit under test. Another clip attaches to ground, providing improved sensitivity and noise immunity.

Probes are ideal for finding short-duration, low-frequency pulses difficult to see on an oscilloscope, but more often they're used to quickly locate gates whose output is hung, or locked, in a high or low condition.

A useful method of circuit analysis with the probe is to start at the center of the suspected circuit and check for the presence of a signal. Using the COMPUTERFACTS schematic and Chapter 2 as a roadmap, move backward or forward toward the failed output, as shown in Fig. 3-11.

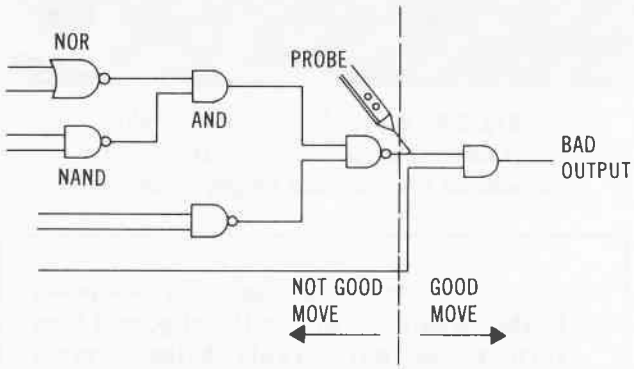


Fig. 3-11. Problem analysis starting at the center of a suspected circuit.

It doesn't take long to find the faulty chip whose output isn't changing.

The only limitation of logic probes is their inability to monitor more than one line.

Logic Pulser

If the circuit under test doesn't have a pulsing or changing signal, you can inject controlled pulses

into the circuit using a logic pulser (Fig. 3-12). These handy devices are portable logic generators.

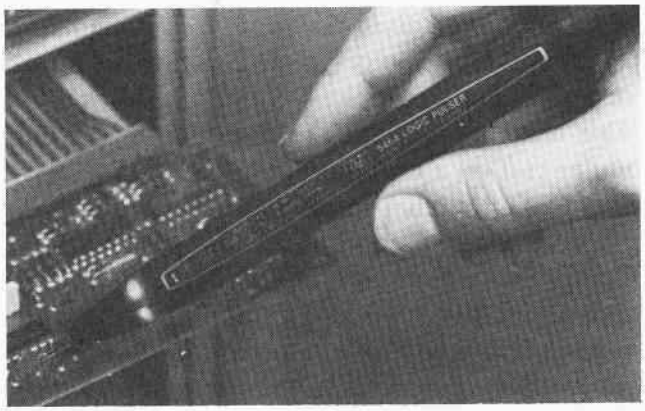


Fig. 3-12. A logic pulser can be used to inject a signal into a circuit. *(Courtesy Hewlett-Packard)*

When activated by a push-button or slide switch, the pulser will sense the logic level at the point touched by the tip and automatically generate a pulse or series of pulses of the opposite logic level. The pulses can be seen on an LED lamp built into the handle of the pulser.

The ability to introduce a changing signal into a circuit without desoldering or cutting wires makes the logic pulser an ideal companion to the logic probe and logic clip. These two tools used together permit step-by-step stimulus/response evaluation of sections of a circuit.

Figure 3-13 shows several ways to test logic gates using the probe and pulser. Assume the output of the NAND gate remains high. Testing inputs 1, 2, and 3, you find them all high. This condition should cause the AND gate output to go high, producing a low out of the NAND gate. Something is wrong. Placing a probe at the AND gate output, you discover the level is low. It should be high. Now, which gate is bad?

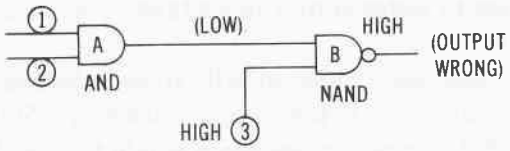


Fig. 3-13. There are several ways to test logic gates like these.

To find out, place the probe on the NAND (gate B) output and the pulser on the AND (gate A) output (NAND gate input), as shown in Fig. 3-14. Pulse this line.

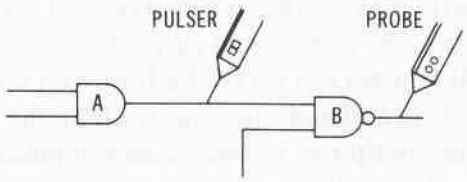


Fig. 3-14. Place the probe on the NAND gate output and the pulser on the AND gate output.

The probe should blink, indicating a change at the input to the NAND. If it doesn't blink a change, the NAND may be bad. But what if the low was caused by a short to ground at the AND output or the NAND input?

Place the probe and the pulser on the AND output trace, as shown in Fig. 3-15, and pulse this line. If the pulse blinks, the NAND is bad; its input changed state, so its output should have changed state also.

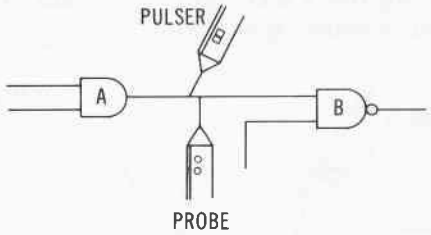


Fig. 3-15. Place the probe and the pulser on the AND gate output.

If the probe doesn't blink, you know this line is shorted to ground. One way you can determine which chip is shorted is by touching the chip case. A shorted chip gets hot, while a chip hung at one level seems to be normal but just won't change state.

Current Tracer

A fourth handy troubleshooting aid is the current tracer probe. This portable device lets you

precisely locate shorts on the computer's system board (or peripheral card). The current tracer senses the magnetic field produced by the flow of electrical current in the circuitry. The logic pulser can be used to generate a pulsing signal that will make the current tracer LED blink, indicating the presence of current.

If you set the tip of the tracer on a printed-circuit line and slide the tracer along the line, an LED in the tip end of the tracer will pulse as long as there is a current present. When you slide past a shorted point, the lamp will go dim or out, and you've found the short.

Figure 3-16 shows an easy way to determine which gate has the short to ground in a logic circuit. Assume gate B has a shorted input. Place the pulser and the tracer midway between the two gates. Adjust the LED in the current tracer so that it just lights. Pulse the line as you place the tracer on the output of A and then on the input to B. The gate with the short to ground will pulse brightly because most of the current is going to ground here. Therefore, the input to B causes the tracer lamp to pulse brightly, while the A side of the line doesn't cause the LED to light. Following the LED light with your tracer will lead you to where the current is going.

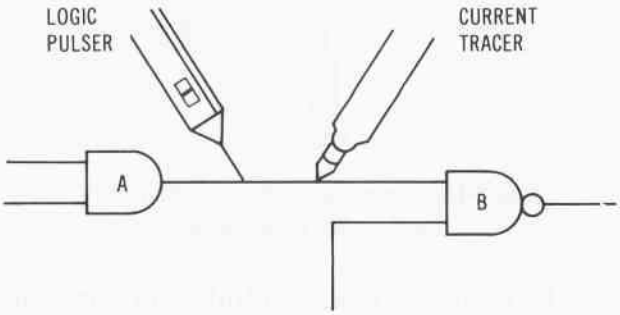


Fig. 3-16. Following the LED light with your tracer will lead you to the short.

IC Testers

Advanced troubleshooting equipment is becoming very sophisticated (and expensive). Today, you can buy equipment that tests almost every chip in your system for between $1000 and $2000. For $10,000 you can even conduct your tests from a remote location.

Micro Sciences, Inc., in Dallas, Texas makes an IC tester that can test over a hundred 7400 TTL and 4000 CMOS series devices. Options for this tester include RAM and ROM tests.

Microtek Lab in Gardena, California makes a tester that can do complete functional pin tests of all 900 devices in the 54/74 TTL series chips. This test tool displays the condition of the chip under test on a liquid crystal display (LCD). It uses LEDs to signal go/no-go test results.

VuData Corporation in San Diego, California markets an in-circuit component tester that's actually a 50 MHz CRT to display the voltage versus current characteristics for virtually all circuit components including capacitors, diodes, integrated circuits, resistors, and transistors. With their tester, the condition of the component under test is determined by the shape of the CRT display. Using this test machine, you can easily pick out open circuits, shorts, leaky diodes, leaky transistors, and marginal ICs. The tool is valuable because it can test a wide selection of components while they're still mounted in the circuit.

Oscilloscope

The oscilloscope has been called the "eyes of the technician." Scopes have been with us for years, although recent advances in the state of the art have added a great many capabilities to the instrument.

Simply put, an oscilloscope (Fig. 3-17) is an electronic display device that draws a graph of signal voltage amplitude versus time or frequency on a CRT screen. By graphically capturing the features of a signal, a scope can be used to analyze the quality and characteristic of an electronic signal. Its interface to the printed circuit board is a probe that touches a test point in a circuit. It can also be used as a measuring device to determine the voltage level of certain signals.

Scopes come in all sizes, shapes, and capabilities. Prices vary between $500 and $20,000. Some scopes use a single test probe for displaying and analyzing a single trace signal. Others have two probes and display two different

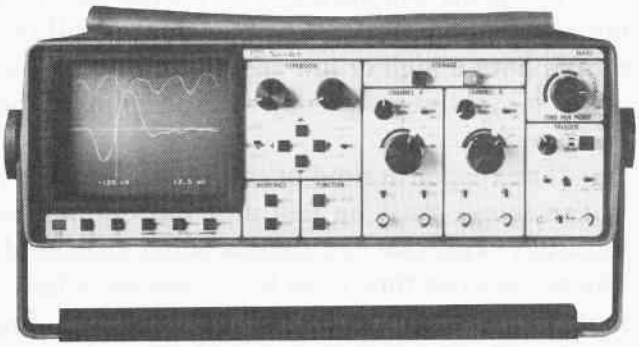


Fig. 3-17. A digital oscilloscope suitable for troubleshooting IBM PC failures. *(Courtesy Nicolet Oscilloscope Division)*

signals (dual trace) at the same time. As many as eight traces can be analyzed at the same time on some oscilloscopes. In fact, in 1983 the first seven-color digital scope was introduced by Test & Measurement Systems Company. Colors make it possible to rapidly compare signals at different locations in the circuitry. Some scopes even have built-in memories to let the machine store a signal of interest for future evaluation.

Besides sensitivity and trace display, one of the major distinguishing characteristics of oscilloscopes is that they allow a great range of frequencies to be observed on the CRT screen as frozen images. This range is called bandwidth. Bandwidths vary between 5 MHz and 300 MHz, and price is proportional to frequency.

Oscilloscopes are useful tools for freezing an analog or varying signal and displaying this static waveform on the face of a CRT screen covered with a measurement grid. While it is time consuming to learn how to use an oscilloscope, the analytical rewards are substantial. Not only can you measure voltage amplitudes and frequencies of test signals, you can also accurately resolve small time intervals and measure delay times, signal rise and fall times, and even locate the intermittent glitch.

Graticule waveform analysis requires careful positioning to help identify the vertical and horizontal intervals of the signal, interpolation of the readings, and calculating the resultant values. Most scopes today have overcome the limitations of graticule measurements by providing a digital readout of the signal based on the position of special screen cursors. These digital scopes have the same basic measurement accuracy of analog scopes with crystal-controlled clocks, but cannot match the analog scope's time resolution and bandwidth (especially for intervals less than 1 microsecond).

One of the most exciting test products introduced recently is the digital storage oscilloscope (DSO). The primary advantage of a DSO is that it can store waveform signals being observed for later analysis and parametric measurements such as rise and fall times, frequency, and period. It can also be used to compare stored waveform images from another unit with the signals present in the unit under test.

Some DSOs can interface with a PC so a service technician can transfer recorded information from a known good machine onto floppy disk magnetic storage media. This capability can also be used to store failure symptoms to develop a magnetic file of past PC failures. Then if a technician is later working on a failed unit, he or she can access a record of past failure symptoms seeking a match for the problem being analyzed. By storing the waveforms from a known good machine and past failures, the DSO/PC configuration can be used to help and stimulate the troubleshooting process.

Most DSOs have more than one available channel. Many standard scopes measure only events that are displaced from the main sweep triggering event. This excludes the trigger event itself from measurement. A scope that includes the main-sweep triggering event in the possible delay range with the run-immediately-after-delay mode may not include the main-sweep triggering event in the trigger-after-delay mode. Most digitizing scopes lack the timing resolution and bandwidth required for analyzing really fast signals. Fortunately, the signals found in the IBM PC circuitry can be easily handled by a DSO.

Of particular interest is the capability of a scope to provide multiple channel analysis at the same instant in time. If you're monitoring an intermittent event and have a suspicion that the problem is related to or caused by another condition, you can use a multichannel DSO

to tackle the malfunction. Putting one set of probes on the test point of the unit under test and another set of probes on the other suspected signal, you can observe both waveforms at the moment the intermittent occurs.

A limitation of the multichannel analysis can be seen when small mismatches in probe- and channel-delay times exceed the allowable error of a measurement. Better quality scopes provide a front panel adjustment to correct for this mismatch. Typically this is called "Channel 2 Delay Matching." Another problem that has occurred using multichannel scopes is that the vertical bandwidth and the transient response errors can obscure the true 50 percent (or other) reference points on the signal displayed.

However, the nice thing about dual trace, quad trace, and even eight trace is the ability to look at different signal paths or different signals simultaneously. For example, you could look at the input and output of a gate and actually see and be able to measure the delay time for the signal passing from input to output of the chip. Another useful technique is simultaneously displaying all or parts of the data bus, or part of the address bus to see what the logic level (high = +5 V, low = 0 V) is and what binary number it represents.

One way to use a dual trace scope to locate the point where a signal ceases erratically is to place the probes at the input and output of a suspected stage. You could also start with the probes several stages apart. Monitor each end of the circuit and start moving the scope probes toward one another until one probe reaches a point where the signal goes haywire. At this point, one probe will be touching the last point of the signal path that has a signal, while the other probe is on a point where some signal amplitude loss occurs at the next point of the signal path. Only one or two components separate the two test points. The problem will be found in the circuitry between the two scope probes, greatly reducing the number of suspected components.

Selecting a scope for the service center is much like buying an automobile. You must first know what you intend to do with the machine. When selecting an oscilloscope, determine what type of signals will normally be measured. Will the signals be repetitive or transient. Will the signals have a high or low repetition rate. What is the maximum frequency of the signals to be measured?

The response time of the vertical circuitry in the scope must be fast and flat enough to faithfully represent the signals being measured. The scope's rise time must be considerably faster than the transition times of the signals being monitored. The step response of the probes and the probe-ground connections should be clean. Coaxial connections may be required to obtain good time measurements.

Once the types of signals to be captured have been determined, you should decide the level of signal detail that must be observed on the screen. This is reflected in the accuracy, amplitude, DC offset, probe effects, range, sensitivity, vertical resolution, and whether single-ended or differential measurements will be made. DC offset is a way to obtain higher vertical resolution without resorting to AC coupling of the trigger signal.

If the signals will be repetitive, real time or equivalent time sampling can be used. A scope's real time scale represents the actual time at which succeeding samples are taken. An equivalent time scale represents the distance between two samples. With this sampling method, the sampling rate must be at least twice that of the highest frequency being measured (the so-called Nyquist rate). Repetitive signals above 100 MHz are better captured using equivalent time sampling.

If the most complex equipment that you'll be troubleshooting is the IBM PC, you can probably get along fine with a dual trace, 25-30 MHz scope. Investment in an oscilloscope is quite cost effective for a service center. A nice scope, recently advertised under $1400, boasts 100 MHz capability, eight trace, two time bases, alternate triggering, three-channel inputs, 0.5 millivolt sensitivity, 0.3 microsecond per division to 0.5 second per division time scale, and five millivolt per division to 5 volts per division voltage scale capability. This is an economical way to guarantee capability to attack the most difficult microcomputer failures.

After you fully understand how the scope works (you've read the owners manual and have followed by tutorial in the reference material) you're ready to make the plunge. When using an oscilloscope to troubleshoot a board for a bad component, the first step to take is to check the scope itself. Make sure the scope and the probes have been recently calibrated. If they aren't, you can't be sure whether the signal you are measuring is accurately represented. Consider powering up the scope and leaving it on for about 30 minutes to let its electronics reach ambient operating temperature. In this way, it won't give you problems that sometimes occur when you try to "read" a circuit in the midst of the warming up process.

Do preliminary checks first (see Chapter 4). Then try to get the problem to loop. Usually, you need the problem to occur during some program loop operation before the failure can be seen on a scope. In some cases the computer is always looping on the problem (for example, video, keyboard, memory refresh, and so forth). In other cases a program can be written that will loop on the problem. In those cases where you can't get the problem to loop, the use of a logic analyzer may be appropriate.

Start checking the control signals to see what is being activated or not activated. Attempt to determine what the PC thinks it's doing.

You need to collect as many clues as possible to help you decide what path through the circuitry you should take. Keep an open mind. Don't look for a particular type failure such as a short or open connection in a component or on the board. Assume that the problem could be any of many things–from a simple short to a capacitance noise problem on the main data bus.

Follow every signal in the circuitry affecting or being affected by the problem. Look for any clue that could provide more insight into the problem and lead you to the failed part, or another clue. Check voltages, data, and control signals. Compare related signals. Also compare signals that are remotely related (for example, data bus signals). If you have another PC system, try to loop on the same thing in that system and compare the measurements from each system.

Many technicians who've worked in this field for a time sometimes feel overwhelmed by the complexity of faults, such as capacitor and resistor failures. Don't become overwhelmed by failures that rarely occur. Capacitors and resistors seldom fail, but, when they do, the failure is usually quite visible. A big chunk blown out of a cap, or a discolored resistor whose value coding you can no longer read are visible indications of a short in part of the circuitry –something shorted with the power on. One cause can be pulling out or plugging in an adapter card while the board is energized. This can also happen if you short leads or traces on the circuit board with a scope probe. Use the utmost care while troubleshooting an energized circuit.

There isn't one scope setting that is appropriate every time you use the scope to troubleshoot PC failures. Many different types of failures can be scoped, and not every one will require the same scope settings.

Logic Analyzers

The logic analyzer is a multichannel oscilloscope with a memory. It captures and stores several digital signals, letting you view the signals simultaneously. If each signal is a bit on the data bus, you can see the entire data bus at one time. This means you can analyze the logic level for each bit on the bus for any instant in time. The bus signals are frozen for your display and analysis. The ability to freeze a single event or data pattern so you can determine the information present on a digital bus during a specific sequence in time is a distinct advantage for troubleshooting.

Logic analyzers, like oscilloscopes, cost between $500 and $20,000, and they come in a range of frequency bandwidths between 2 MHz and 200 MHz.

These analyzers can display many signals (channels of input) simultaneously. Arium Corporation in Anaheim, California sells an analyzer that handles 32 channels of input data at frequencies up to 100 MHz. Nicolet offers a 48

channel, 200 MHz analyzer with built-in microcomputer and dual, double density floppy disk drives. Each channel has associated with it a probe clip for connecting to some test point in the circuitry. Fortunately, the clip probes are tiny and easy to install.

A sampling of the capabilities available in logic analyzers reveals one configuration that provides 104 channels at 25 MHz, another with 32 channels at 100 MHz, and yet another with 16 channels at 330 MHz. Another configuration has 8 channels of input and can operate at 600 MHz! (Recall that your IBM PC clock is 4.77 MHz.)

Where would logic analyzers be useful? One place is in debugging software. You can read the data in machine code and trace its flow through the circuit. You could analyze the input and output of each memory bit on the data bus simultaneously for locating a bad RAM chip. Or you could uncover intermittent glitches, those phantom spikes that can raise havoc with your system. There are many more uses for logic analyzers, including analysis of disk I/O operation.

There are advantages and disadvantages associated with using a logic analyzer. One advantage is that you needn't loop on the problem. The analyzer will develop a historical visual record of the signals occurring at the points being measured. Logic analyzers are also great for intermittent failures or heat problems. However, this equipment is very expensive, doesn't measure voltage levels, is time consuming, and is difficult to connect to the circuit under test.

When troubleshooting with a logic analyzer, the first step is to determine what to analyze. You must look at the symptoms and decide which signals to check. If you've already checked the power and timing, and are ready to apply a logic analyzer to the problem, begin by connecting the analyzer probe clips to the control signal pins in the suspected area of circuitry. Connect the remaining probe clips to signals that are affected by the control signals being monitored.

Often technicians will get overwhelmed investigating problems on the large address and data buses. If you take the time to learn how to read these buses, you'll find them valuable when analyzing system board problems. With a logic analyzer you can visually observe conditions up to and during the time of failure. This can be done by connecting the analyzer to the data bus and the address bus at a point where they are active into the RAM memory. In this configuration, you can "read" exactly what's going on in the system's "mind." To do this, you'll need a logic analyzer with at least 32 channels. Then you can simultaneously monitor both the 8-bit data bus and the 16-bit address bus, and still have enough probes left to trigger on other signals.

You must understand the operation of the analyzer so you can set proper triggering and be able to recognize the screen display. Read the logic analyzer's operating manual to become comfortable with the tester. Logic analyzers can be a blessing in disguise, and they can help isolate almost any problem. But when hooking up 32 channels of tiny probe clips, there is ample opportunity for mistakes.

To save time and make troubleshooting easier, you can build a kludge that lets you connect all the necessary leads from the analyzer's pod to several regular chip clips at one time. By plugging the clips to the appropriate IC (labelled on each chip), with the correct pin 1 alignment, you can have the address and data lines ready for analyzing in less than 2 minutes. Individual clips can take up to 15 minutes to connect without the kludge. Several magazine articles have described how to construct such a kludge.

The logic analyzer has been called the oscilloscope of the digital domain. It can be a valuable tool for the software or hardware designer, but the investment can be large. Fortunately, most IBM PC failures can be found and corrected without the need for a logic analyzer.

Signature Analyzer

Logic probes can be effective in detecting logic levels and pulses at single points. Oscilloscopes can extend the number of points to be monitored even though the data pulses all tend to look alike.

And logic analyzers extend the number of test points even further to include buses the size of the data and address buses. However, as the sophistication and capability of the measurement device increases, so does the expertise required to operate the test tool. Logic analyzers, in particular, can be very capable but they can also be difficult to understand and operate. The signature analyzer was developed to allow easy detection of hardware failures.

Signature analysis is a comparison method of troubleshooting. It works by running a diagnostic program in the system being tested, and evaluating a coded signal at specific test points in the circuitry. If the coded signal matches the code observed when the system was running properly, the malfunction is not in that part of the circuitry. When a test point signature fails to match the baseline correct code, this suggests that you have located the faulty area. Then you can probe backwards or forwards from this point to isolate and locate the component that has failed.

The key to the success of this test technique is in the signature code. The first codes were developed by Hewlett-Packard and with slight modification are still being used today. A test code is a 16- or 24-bit repeatable value that represents a stream of data passing a test point during an interval of time. This known stream of data, when sampled at different places on a good circuit board by a signature analyzer, produces a unique 16- or 24-bit code at each test point.

These codes can be documented or stored in a programmable read only memory (PROM) and recalled later for comparison during troubleshooting. The PROM then becomes a custom memory module containing every signature sampled from a properly working system that was being stimulated or pulsed with a known data stream.

Signature analysis has not been a popular troubleshooting tool because it takes lots of time to identify the test points or nodes, probe the nodes, produce a signature, and then document the code. Once this task is completed, however, the task of locating a failure becomes a breeze. And the introduction of PROM modules has made the setup task much easier.

More improvements in this analysis technique can be expected soon. One analyzer on the market uses a mode called "backtrace" to prompt the troubleshooter through a series of test points, guiding the tester to trace bad signatures back to the failed part.

The investment for a signature analyzer is between $400 and $10,000. Signature analysis uses a simple, nontechnical approach to troubleshooting, so even untrained people can use the equipment and the technique.

Now that you understand what kinds of tools are available, let's look at how you can use these tools to find failed components.

USING TOOLS TO FIND FAILED COMPONENTS

Most components on the IBM PC motherboard are TTL (transistor-transistor logic) chips. If you know the logic gates in a chip to be tested (NAND, NOR, OR, AND, and so forth) you can test for opens or shorts by applying a known logic level to the inputs while monitoring the output. For example, if you were to place a slowly pulsing 0 volt to +5 volt signal on the input to the AND gate in Fig. 3-18 with both inputs shorted, you should see the output voltage level change (pulse) along with the input. Whenever the input is a logic high, the output becomes a logic high (between +5 volts and +2.4 volts).

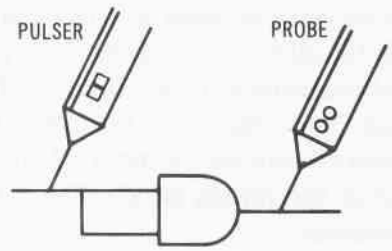


Fig. 3-18. Pulsing the input should cause a change in the output as indicated by the logic probe.

The tool you use on the input is a logic pulser. The monitor tool on the output is a logic probe. The pulser places a cyclical logic level on the input to the device and the probe measures

the presence or absence of a logic signal on the output of the chip.

If the input to the AND gate becomes shorted to ground, the pulser cannot cause the gate to react to its signal and the output remains at a logic 0, or low (about 0 volt). Even if just one of the inputs shorts to ground, the output cannot change and remains low.

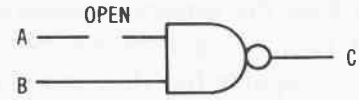
A short to the gate supply voltage (+5 volts) will have the effect of qualifying or enabling one input to the gate all the time. This means that each time the other input receives a logic high, the input set is correct and causes the output to change to a logic high even though only one input signal was actually correct. This produces incorrect circuit operation and strange results. This kind of problem shows up in memory circuits. Only one of the inputs to a particular gate is shorted or opened. Whenever this gate is used, the resulting output may be correct–a difficult problem to trace down.

Shorting an input pin to +5 volts can have potentially disastrous results. When the previous gate tries to deliver a logic 0, or low, a huge current is produced which usually causes catastrophic failure of the driving chip. The same result occurs when the input pin is shorted to ground and the previous gate tries to deliver a logic 1, or high. The +5-volt logic high is shorted directly to ground, producing an unusually high current with equally disastrous results.

Open connections prevent logic levels from being transferred and prevent the affected gate from being able to respond. If one input of a two-input NAND gate is open at the input as shown in Fig. 3-19, all but one of the four possible input combinations will be correct. This means that with this type of failure, the system could operate correctly most of the time with only half of the inputs good. The failure would be intermittent.

As shown in Fig. 3-20, if the device being tested is a NOR logic gate, the output would be a logic 1 only when both inputs are at logic 0. Should one of the inputs become open, it would float to logic 1 and cause none of the input conditions to produce a logic 1 output. Thus, the output would be low all the time–just as though the output were shorted to ground.

If the chip has an open pin at its output, it cannot deliver any logic 1 or 0 to the next gate.



INPUT A	INPUT B	OUTPUT C
L	L	GOOD
L	H	BAD
H	L	GOOD
H	H	GOOD

Fig. 3-19. An open at the input to a NAND gate is only a problem in one of four logic state cases.

INPUT A	INPUT B	OUTPUT C
L	L	L
L	H	L
H	L	L
H	H	L

Fig. 3-20. An open at the input to a NOR gate will prevent the output from ever changing state or going high.

You can measure a voltage at the input to the next gate since it is providing the potential, a logic 1 or high level (something around +4 volts). The key here is that any time an input to a TTL gate opens (a condition we call "floating"), the gate will act as though a logic 1 were constantly applied to that input. The voltage on this floating input will drift between the high supply voltage of +5 volts and a level (about 1.5 volts) somewhere between a valid high and a valid low. (A valid high is usually above +2.4 volts; a valid low is below +0.4 volt.)

A voltmeter reading of about +1.7 volts at the output pin of a gate on a chip is a clue that the output is floating open and the voltage is actually being provided by the next chip or following gate.

All these kinds of failures can be located using a logic pulser and a logic probe with backup from a VTVM for voltage measurement.

Since the PC system board is flexible at certain points, replacing chips by depressing the board without supporting it from beneath could cause a break to occur opening a trace on the circuit board. A hairline crack such as this is often difficult to find, but looking at the board with a magnifying glass and a strong light (or a

magnifying lamp) can sometimes reveal a suspected failure. A resistance test can be conducted with a VOM or VTVM by placing a probe at either side of the suspected bad trace as shown in Fig. 3-21 and observing whether a zero ohm reading is measured. Another way to ascertain if an open trace is present is to compare the logic states at either end of the trace.

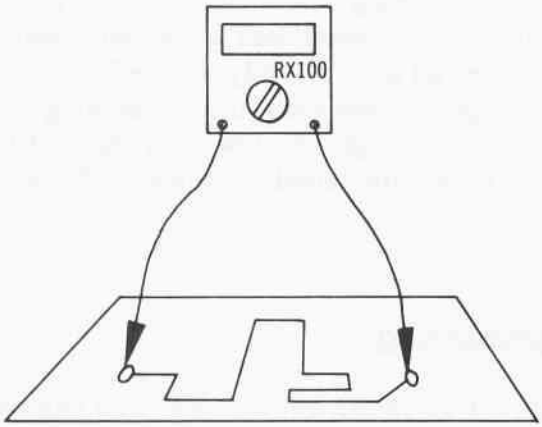


Fig. 3-21. A trace can be tested for an open using an ohmmeter to test the logic states at either end of the trace.

An important fact to keep in mind when testing for individual shorted or open gates in the IBM PC system board circuitry is that more than one gate may use the same input or output lines to or from another gate. This is called "fan-in" or "fan-out." When studying the gate circuitry, remember, the failure could be located at the other end of the board. One long trace from it to the chip you are looking at may be shorted or open at that end. Use of the schematics in Chapter 2 and the SAMS COMPUTERFACT for IBM PC will be of value here.

OTHER TROUBLE-SHOOTING TECHNIQUES

There are some interesting tricks you can use to aid in finding chip failures.

Use Your Senses

Look, smell, and feel. Sometimes failed components become discolored or develop bubbles or charred spots. Blown devices can produce some distinctive smells–the smell of a ruptured electrolytic capacitor, for example. Finally, shorted chips can get very hot. By using a "calibrated finger," you can pick out the hot spots on your board.

Heat It, Cool It

Heating and cooling is a fast technique for locating the cause of intermittent failures. Frequently, as an aging device warms up under normal operation, it becomes marginal and then intermittently quits working. If you heat the energized area where a suspected bad chip is located until the intermittent failures begin, and then methodically cool each device with a short blast of canned coolant spray, you can quickly cause a marginally defective chip to function again. By alternately heating, cooling, heating, and cooling, you can pinpoint the trouble in short order.

You can heat the area with a hair dryer or a focused warm air blower designed for electronic testing. Be careful using this technique since the thermal stress you place on the chips being tested can shorten the life of good components. A 1- or 2-second spray of freeze coolant is all you should ever need to get a heat-sensitive component working again.

Most coolant sprays come with a focus applicator tube–use this to pinpoint the spray. Avoid spraying electrolytic capacitors, because the spray soaks into the cap, destroying the electrolyte in some aluminum capacitors. Also be careful not to spray your own skin. You could get a severe frost burn.

The problem with hot air and freeze spray techniques is that the heating and cooling can weaken good chips. A logic analyzer can replace temperature stress techniques to find "weak" components by triggering on the failure.

Some technicians use both test methods together. In this technique, the analyzer is connected to the suspected circuit and a suspected IC is heated (if the system is temporarily working), to cause a failure, or cooled (if the system has already failed) until the circuitry starts working again.

Each part is not heated or cooled until the failure occurs because it may not be the bad component and you may force a new failure. Only the part that is suspected bad should be tested this way.

Assume that you are searching for a heat sensitive component. Your visual and preliminary checks don't turn up a thing. You analyze the symptoms looking for clues that might lead to the area of failure. The program runs and then the system locks up. So you open the equipment. You suspect a bad RAM IC on the system board. After all, memory chips fail often, don't they? So you spray coolant on each of the RAMs starting with U69. While spraying U72, the system suddenly starts working again. Thinking you've found the bad part, you replace U72. Buttoning the system back up, you rerun the program that was failing earlier. Ten minutes later, it locks up. Notice that, if you had buttoned the system up and only tested for one or two minutes, you would have assumed the computer repair complete and placed the system back in service. Without a 30-minute operational test to confirm problem correction, you may find yourself embarrassed and losing credibility fast when you're called on later to "fix that darn thing" again.

The retest failed. If you assume that this is one of those rare double chip failures, you reopen the unit and start spraying coolant on the RAMs again. This time, just as you are about to spray RAM IC U89, the system starts up again. Being a "thorough" technician, you let the system lock up again, and then you spray U88 again (this was the last chip you sprayed before moving to U89). Bingo! The system starts to run. Smiling with confidence that you've found the culprit component, you replace U88 and rerun the program. To your dismay, the machine locks up again! Irritated and frustrated, you start wildly spraying again. After an hour and a half (and three cans of coolant), you find the problem is the data buffer U12. The data buffer not a RAM! If you had used a logic analyzer to monitor the address and data buses while operating the PC to failure and cooling the system back into operation, you would have seen the failure and could quickly follow it to the bad buffer chip. Many technicians have used coolant spray hoping to identify heat-sensitive components only to replace a part that seemed to be helped by the spray and discover the problem unchanged.

The important thing to remember here is that when troubleshooting any failed part or failing circuitry, always keep your mind and eyes open for anything that could represent a clue to the problem. Sometimes a clue will lead you back over an area already covered. Don't be discouraged. Remember, troubleshooting can be detailed and tough. Follow each clue until you finally isolate the failed component. Never give up.

Piggybacking

Figure 3-22 shows another way you can chase down intermittents caused by a break (open) in a chip bond (wire) inside the chip housing that allows good contact only when the chip is cool.

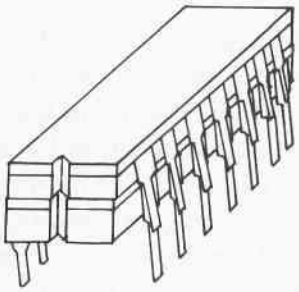


Fig. 3-22. Open-type failures can be found by piggybacking a good IC on a suspect IC.

Place a good chip over the top of a suspected chip and energize the circuitry. You may need to squeeze the pins of the good chip in slightly so they make good contact with the pins of the suspected device.

If the intermittent failure is caused by an open connection, the new chip will react to input data and cause its output to act accordingly. Use your stock of spare parts as your piggyback source. A major caveat here is that if the failure was caused by a short on one of an IC's pins, piggybacking a good chip over the shorted chip will blow out the good chip. This technique must be used with caution.

The Easter Egg Approach

Often we can quickly locate a fault to a couple of chips but need further testing to determine which chip is the culprit.

When time is of essence, take an "Easter Egg" approach. Just as a youngster used to pick up and examine Easter eggs one at a time to see if its name was marked on the egg, you can try replacing the chips one at a time to determine if the chip replaced was causing the problem. You have a 50-50 chance of selecting the right chip the first time. If it didn't work, replace the other chip.

If the chips involved are inexpensive "jelly beans" (7400 series TTL), why not replace them both? For 30 cents more, go ahead and splurge. If the problem's gone, but you're still curious, you can always go back later and test each chip individually.

MICROVOLT MEASURING A PIECE OF WIRE

If you have a meter with microvolt sensitivity and have isolated a "stuck low" problem to two chips, you can try the technique shown in Fig. 3-23. Measure the voltage drop between input pin 1 of gate B and output pin 3 of gate A.

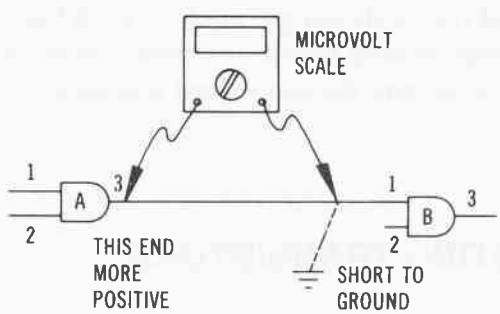


Fig. 3-23. A short circuit that is sinking current can be found using a meter with microvolt sensitivity.

This means measuring the opposite ends of the same trace or piece of wire! You're interested in determining which end of that trace is the more negative. The end nearest a bad chip will be more negative because the defective chip will short the trace voltage to ground causing a smaller microvolt reading at this point.

TESTING CAPACITORS

How do you check out a capacitor that you believe has failed? If the device has shorted, resulting in severe leakage of current, you can spot this easily by placing an ohmmeter across the capacitor and reading the resistance. At first you'll notice a low reading, because the capacitor acts as a short until it charges; but then, if the capacitor is working properly, it will charge, and the resistance will rise to a nominally high value. If the device is shorted, the initial low resistance reading continues and the capacitor won't charge.

Should the component be open, you'll not see the instantaneous short at T0, the moment charge starts to build. An open circuit has infinite resistance. An in-circuit capacitance tester is helpful here.

Total failure as a short or open is easy to find. But how about the device which its leakage depends on temperature or its dielectric has weakened, changing the capacitance value? To test this capacitor requires a different level of analysis.

CAPACITANCE MEASURING

If you have an ohmmeter which has the number 10 in the middle of the scale, you can easily use it to approximate the capacitance of a device. Using the time constant formula T = RC, where the time in seconds for a capacitor to charge to 63.2% of supply voltage is equal to the resistance in ohms times the capacitance in farads. Using a 22 microfarad (or 0.000022 farad) capacitor and a 1 megohm (1,000,000 ohm) resistor, the charge time for one time constant is 0.000022 x 1,000,000 = 22 seconds.

Transposing the formula to read:

$$C = \frac{T}{R}$$

where

C is capacitance in farads,
T is time in seconds,
R is resistance in ohms.

you can determine the value of capacitance by knowing the resistance and counting the seconds required for the charge to cause the ohmmeter needle to reach 63.2 percent of full scale (infinite resistance). This point is at about 17 on the meter's scale.

To do this, disconnect one end of a capacitor from the circuit, turn on your meter, and let it warm up for a minute. Zero adjust the ohms scale reading. Then estimate the ohms scale multiplier needed to let the capacitor charge in some acceptable time period. For microfarad capacitors use the x 100K scale because this will let the capacitor charge in less than a minute. The 17 on the scale represents 1.7 M ohms on the x 100K scale.

Short a low-ohm-value resistor across the two capacitor leads for several seconds to thoroughly drain off any charge. Then connect the meter's ground lead to the negative side of the capacitor (either side if the capacitor is not an electrolytic), and touch the positive meter probe to the other side of the capacitor. Using a stop watch to count seconds and tenths of a second, watch the face of the ohmmeter as the capacitor charges, moving the resistance needle up. When the needle gets to 17 on the scale, stop the clock and read the time. This will give you the value of capacitance in microfarads.

This technique will give you a close enough approximation of the capacitance value to determine if the device is good or should be replaced.

REPLACING CAPACITORS

Always try to use the same type and value capacitor as the one being replaced. Keep the leads as short as possible and solder the capacitor into the solder connector holes with the proper iron. The solder process should not exceed 1.5 seconds per lead, heat damage to the component may result.

A good technique to use is to tin the capacitor leads just before poking them through the circuit board holes. This speeds the solder bond process.

TESTING DIODES

If you have a digital multimeter (DMM) with a diode test capability, you can quickly determine whether a suspected diode is bad or good. Placing the meter on the ohmmeter setting and the probes across the diode causes the meter to apply a low current through the diode if the diode is forward biased. The voltage drop across a diode is normally 0.2 to 0.3 volt for germanium diodes and 0.6 to 0.7 volt for silicon diodes. Reversing the leads should result in no current flow, so a higher voltage reading should be observed. A low voltage reading when biasing the diode in either direction suggests that the device is leaking or shorted. A high voltage reading in both directions suggests the diode bond has opened. In either case, replace the diode immediately.

In-circuit tests of diodes can also be done using the ohmmeter to check the resistance across a diode in both directions. With one polarity of the meter probes, you should get a reading different from that obtained when the probes are reversed–not just a few ohms different but several hundred ohms different. For example, in the forward-biased direction, you could read 50 to 80 ohms; in the reverse biased direction, 300K ohms. This difference in readings is called DE for "diode effect" and is useful for evaluating transistors. When diode readings in both directions show low resistance, you can be sure the leaky short is present.

TESTING TRANSISTORS

It's no fun to desolder a transistor to test it for failure and find that it tests good. You then have to solder it or a new device back into the circuit board and go to the next suspected transistor and desolder it, and so forth.

Fortunately, there is a way to determine the quality of silicon transistors without removing them from the circuit. In 90 percent of the tests, this procedure will accurately determine whether a device is bad.

Figure 3-24 shows that transistors operate the same way as a configuration of diodes. PNP and NPN transistors have opposite-facing diodes. The transistor functions by biasing certain pins and applying a signal to one of the leads (usually base) while taking an output off the collector or emitter.

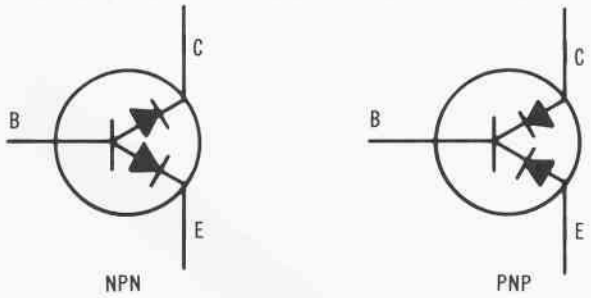


Fig. 3-24. A transistor acts like a pair of diodes.

These tests apply to both PNP and NPN transistors. If an ohmmeter is placed between the collector and emitter as shown in Fig. 3-25, it effectively bridges a two-diode combination in which the diodes are opposing. You should get a high resistance reading with the leads applied both ways. (It's possible to wire the transistor in a circuit which makes the transistor collector-emitter junction act like a single diode. In this case you could get the diode effect. Both results are normal.)

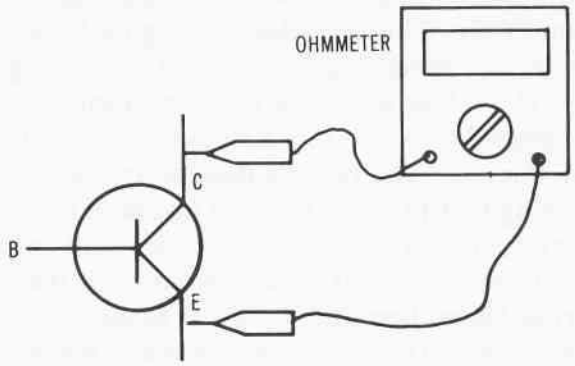


Fig. 3-25. A transistor can be tested using an ohmmeter placed across the collector-to-emitter junction.

Typical collector to emitter resistance readings for germanium transistors are as follows:

Forward biased = 80 ohms
Reverse biased = 8000 ohms (8K)

For silicon transistors you might read:

Forward biased = 22 megohms
Reverse biased = 190 megohms

The high/low ratio is evident and is about the same for both types.

Place the probes across the collector-to-base junction leads. Reverse the probes. You should observe a low reading in one case and a high reading with the test probe leads reversed (the diode effect).

Try the same technique on the base-to-emitter junction leads. Look for the diode effect (Fig. 3-26). If the diode effect is not present in all the previous steps, you can be certain the transistor is bad and needs replacing.

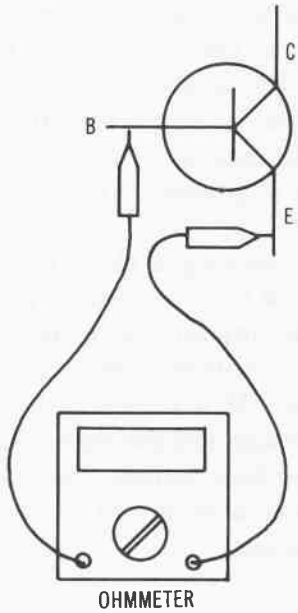


Fig. 3-26. Check the base-to-emitter junction for the diode effect.

Another way to evaluate a transistor is to measure the bias voltage from base to emitter on an energized circuit. Confirm the correct supply voltage first; power supply problems have been known to trick troubleshooters into thinking a certain component has failed.

The base-to-emitter forward bias for silicon transistors should be between 0.6 and 0.7 volt

DC. If the reading is below 0.5 volt, replace the diode–the diode junction is leaking too much current (Table 3-3). If the reading is almost 1 volt, the junction is probably open and again the device should be replaced.

Table 3-3. Base to Emitter Voltage Readings and Action to Take

Base to Emitter Voltage *(forward biased)*	Action
0.5	Replace
0.6-0.7 V	Good, keep
0.9	Replace

Although in some isolated cases some other failure could cause the low reading, the most common cause of low bias voltage is failure in the transistor itself.

If the previous tests are inconclusive, there is something else you can try. Measure the voltage across the collector-to-emitter junction. If the reading is the same as the source supply (+5 volts for Q1 on the color/graphics adapter) and you notice on the schematic that there's plenty of resistance in the collector/base circuit, the junction is probably open. Replace the device.

If your reading is close to 0 volt, take a small length of wire and short the base to the emitter, removing all the transistor bias. The collector to emitter meter reading should instantly rise. If it doesn't, the transistor is shorting internally and should be replaced. If the collector to emitter voltage does rise, it suggests a failure in the bias circuitry, perhaps a leaky coupling capacitor.

SOLDERING AND DESOLDERING TECHNIQUES

Removing Solder

Removing solder from the printed-circuit boards in the IBM PC must be done rapidly. Typically, remelted solder on a joint must be removed in less than 3 seconds. There are many ways to remove solder from printed-circuit boards. One way to remove residual solder is to use a "solder sucker"–a hand held vacuum pump with a spring-driven plunger to pull the hot, melted solder off a connector (Fig. 3-27). The process involves heating the old solder until it melts, placing the spring-propelled vacuum pump in the hot solder, then quickly removing the soldering iron while releasing the vacuum pump's spring, sucking the solder up into a storage chamber in the pump.

Fig. 3-27. The spring-driven plunger in the solder vacuum pump is used to pull hot solder off a connection.

Continuous vacuum solder extraction is another technique for desoldering components from PCBs. This type desoldering tool uses an electrically generated vacuum and tip heating to melt the solder around a joint and rapidly suck the melted solder up into a storage chamber built into the tool. The vacuum desoldering tip has an opening that fits over the lead or pin to produce maximum heat transfer to the joint while providing sufficient space around the lead to permit the molten solder and air to pass through the space. The tip is the primary means to transfer the heat to melt the solder joint. It is also the path through which molten solder will be sucked by vacuum action. A good tip will pull ambient air through the tip to cool the joint and prevent resweating. Resweating occurs when the lead reconnects to the side walls of a plated-through hole in a printed-circuit board.

The tip is also used to manipulate the lead during the desoldering process. Once melting action begins with the tip over the lead, it is used to gently wiggle the lead until the lead swings freely and easily, suggesting full solder melt in the joint hole. Wiggling the lead with the tip provides a mechanical sensation that sufficient solder melting has occurred.

Once the solder has melted, the vacuum is applied while continuing to wiggle the lead. This removes almost all molten solder. It also cools the lead, and the pad, and prevents resweating. Because the similar controlled pressure and heat used to bond a part into a board is used to remove the same part, care must be exercised to prevent delamination. Excessive pressure when applying a soldering or desoldering tool to a solder joint can delaminate circuit pads or runs.

A good rule of thumb is to apply tip temperature heat between 575 and 600° F for about 2 seconds following initial solder fillet melt on the lead side. Then apply vacuum (while wiggling the lead) for 1 or 2 seconds. Once the lead moves freely, full solder melt has occurred. Maintain the lead motion and apply the vacuum suction. With the vacuum still on, gently remove the tip. When the tip is away from the lead, turn the vacuum off.

Lead resweating to the side walls can be prevented by moving the tip during heat and vacuum application. Move the lead in a circular motion when desoldering round leads. Move the lead back and forth along the flat plane when desoldering flat leads.

When desoldering DIP packages, space the pin desoldering sequence to prevent working on adjacent leads and causing localized heat buildup. Do the corners first, then skip over every other pin on the first desoldering pass. On the second pass, desolder the alternate leads.

If sweat joints still occur after desoldering the leads on a component, let the joints cool down. Then resolder the joints that were not completely desoldered and conduct vacuum desoldering once again.

This technique works fine until you try to use it around CMOS chips. Some vacuum pumps produce static electricity, and by now you know what that can do to an MOS or CMOS chip.

A safer way to remove solder is to use solder braid. Solder braid is an inexpensive alternative to vacuum desoldering. In fact, it prevents human error by applying heat too long because the soldering iron never touches the solder. When using solder braid, touch the solder with the end of a strip of braided copper and then heat the braid just a short distance from the solder (Fig. 3-28). The copper braid heats quickly, transferring the heat to the solder, which melts and is drawn into the braid by capillary action. Then, remove the solder-filled braid and cut off the silver solder-soaked portion of the braid and throw it away. The copper-colored portion of the braid is ready for the next application.

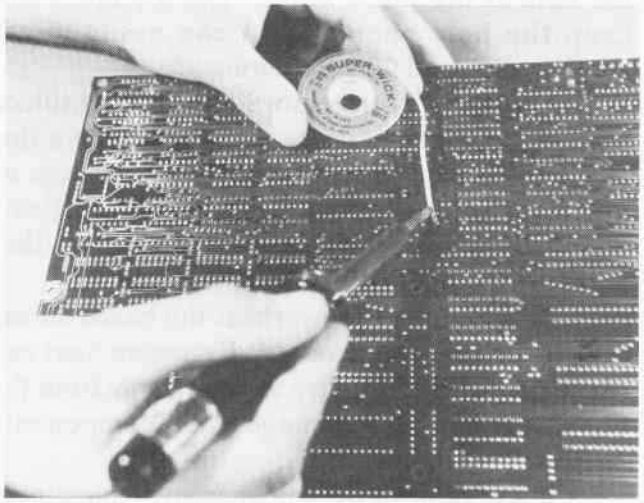


Fig. 3-28. Solder can also be removed from a connection by using a solder wick to draw the solder up into the braid.

Because the soldering iron never touches the solder itself, component traces and the circuit board are protected from thermal damage. This braid can also be used to remove solder bridges or icicles of solder that cause shorts on the board. Desoldering braid works equally well for removing excess solder from plated through holes in printed-circuit boards. The capillary action of the solder wick will pull molten solder easily out from the space between the lead and the hole.

Desoldering braid can be purchased in varying widths ranging from 0.035 to 0.220 inch. The proper size of braid will equal, or be slightly larger than the connection being desoldered.

According to Carolyn Watson in her article "Desoldering today's circuit components" (Electronic Servicing & Technology, October 1984), the smallest size (0.035 inch wide) is good for microcircuits; small pads required braid 0.06 inch wide. Medium pads need 0.075 inch braid, and large pads require 0.11 inch braid. Wicks that are too large may soak up too much heat from the iron restricting heat transfer to the soldered joint. Too narrow a wick will hamper sufficient solder removal and prompt several applications to complete a desoldering operation.

Should you be using a solder removing technique other than copper braid and discover that some solder remains in the circuit board hole, heat the solder and push a toothpick into the hole as the solder cools. The toothpick will keep the hole open so you can easily insert another wire lead for resoldering.

Another way to remove the residual solder blocking a hole is to drill out the hole with a tiny drill bit. Be sure to remove any debris, filings, or pieces of solder before energizing the circuit board. Use a magnifying glass to confirm that nothing unwanted remains on the board.

Be careful not to overheat the board during the solder-removal process. Excessive heat can cause part of the circuitry to come away from the board. It can also damage good components nearby.

If you remove the solder from a component and a lead is still stuck on some residual solder, take a pair of needle-nose pliers and pinch the lead as you gently wiggle it to break it loose from the solder bond. Or you can remelt the residual solder and gently wiggle the lead free.

The pins of some chips are bonded to the circuit board by a process called wave soldering. Wave soldering produces an exceptionally good bond without the added manufacturing expense of a socket. This process helps keep the fabrication costs down, but it makes it more difficult for you to replace the chip.

One effective way to remove wave-soldered chips is to cut the chip leads or pins on the component side and remove the bad chip. Then remove the pieces of pin sticking through the board using a soldering iron and solder braid or a vacuum pump.

Some special tools are available to help you in removing soldered components. Figure 3-29 shows a desoldering tip that fits over all the leads of a chip socket or dual in-line package (DIP) socket.

Figure 3-30 shows a spring-loaded DIP extractor tool. By attaching this device to the chip and then applying the DIP tip shown in Fig. 3-29 to the soldered connections on the opposite side of the board, you can easily remove a complete chip all at one time. Press the load button downward and engage the clips, causing the extractor to place an upward spring pressure on the chip. When the solder on the reverse side melts enough, the chip will pop up and off the board.

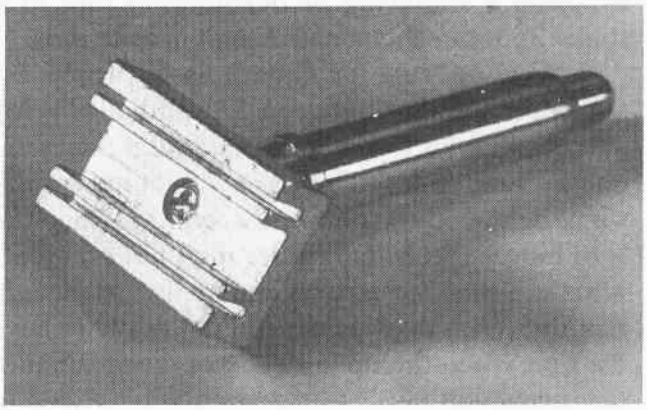


Fig. 3-29. A desoldering tip for removing chips that are soldered to the circuit board.

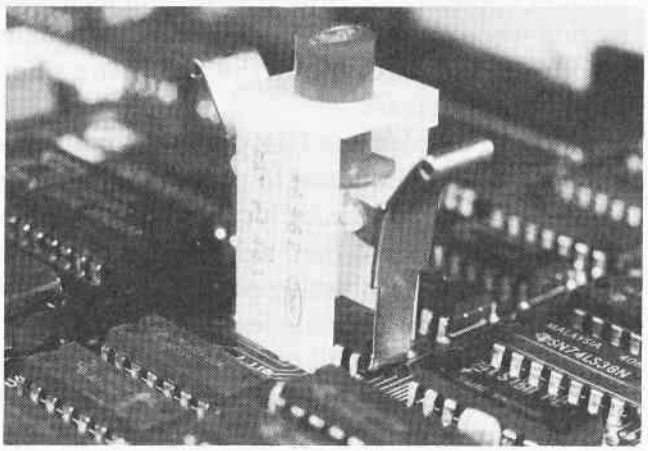


Fig. 3-30. A spring-loaded dual in-line extractor tool.

When you replace a chip that was soldered to the printed circuit board, always solder in a socket and then plug the replacement chip into this socket. This will make future replacements a

lot easier. Be careful to maintain the correct pin 1 alignment.

Soldering Tips

No pun intended. Hand soldering is the most misunderstood and most often abused function in electronics repair. Not only do many people use poor soldering techniques, but they also use the wrong soldering irons.

Solder isn't simply an adhesive bandage making two metals stick together. It melts and combines with the metals to form a consistent electrical and mechanical connection. Time and temperature are critical in this process. The typical hand solder job should be accomplished in 1.5 seconds or less if the soldering iron and tip are properly selected and then properly maintained.

The nominal solder melting temperature is 361° F. Metal combination between the solder and the metals being joined occurs at temperatures between 500 and 600° F.

Most soldering jobs join the metals copper and tin, but both of these metals are easily oxidized. Poor or no solder connections are made if the surfaces to be connected are covered by contaminants such as oils, dirt, or even smog, so be sure to use solder with a good cleaning flux. The flux prepares the surfaces for best solder metalization. The flux melts first and flows over the metal surfaces removing oxidation and other contaminants. Then the metal heats so that the solder melts and flows producing a good, shallow bond.

The key to successful soldering is in the soldering iron tip. Most technicians selecting their first soldering iron jump right into a low wattage iron, but this is a mistake. Instead, pick an iron whose tip operating temperature is suited for the circuit board you're to repair. If the tip temperature is too low, the tip sticks to the surface being soldered. If it's too high, it damages the board surface. The ideal working temperature for soldering on your computer's circuit board is between 600 and 700° F.

The soldering iron tip is used to transfer the heat generated in the iron out to the soldering surface. The iron should heat the tip quickly, and the tip should be as large as possible yet slightly smaller than any soldering pad on the board.

Tips and the contacts or pins that you're soldering are made of the same copper material. Copper quickly conducts heat, but it dissolves in contact with tin. Solder is made of tin and lead. To keep the tin from destroying the copper tip, manufacturers plate a thin layer of iron over the tip. The hot iron (now you know where the term "iron" came from) still melts the solder, but now the tip lasts longer. The iron melts above 820° F, so if the heat produced by the iron stays below 700° F, the solder melts but not the iron plating.

The disadvantages of the iron plating are that it doesn't conduct heat as well as copper, and it oxidizes rapidly. To counteract this, you can melt a thin coat of solder over the tip. This is called "tinning." This solder layer helps the soldering iron heat quickly and also prevents oxidation.

The tip of an old soldering iron is usually black or dirty-brown with oxidation, so it doesn't conduct heat very well. These "burned-out" tips can be cleaned with fine emery cloth and then can be retinned and used.

Wiping the hot tip with a wet sponge just before returning the iron to its holder is a mistake. This removes the protective coating, exposing the tip surface to atmospheric oxidation. It's much better to add some fresh solder to the tip instead. Keep your iron well tinned.

Figure 3-31 shows the proper way to solder a socket or connector lead. Place the tip of the iron on one side of the lead and the solder on the other side.

As the solder pad heats, the tin/lead solder melts and flows evenly over the wire and the pad. Keep the solder shallow and even. When you think your soldering job is complete, carefully inspect your work. Sometimes, if you aren't careful, you can put too much solder on the joint such that there's not enough solder on the top or bottom of the connection. It's also possible to get internal voids or hollow places inside the solder joint. Large solder balls or mounds invite "cold solder joints" where contact is only par-

tially made. Figure 3-32 shows some examples of inadequate soldering. These types of solder joints can be a source of intermittent failure.

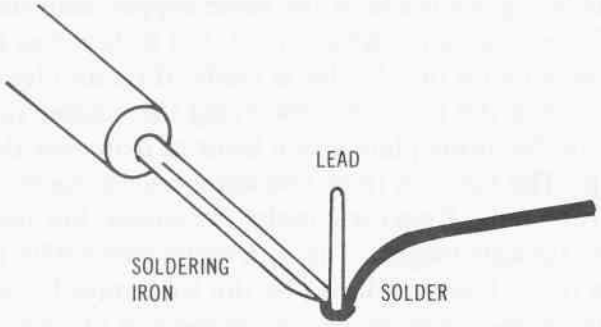


Fig. 3-31. Place the soldering iron on the opposite side of the lead from the solder.

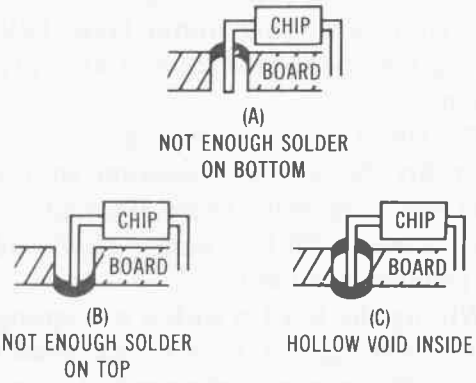


Fig. 3-32. Examples of inadequate soldering on a printed circuit board.

Good soldering takes patience, knowledge, and the right tools–a temperature-controlled soldering iron whose tip temperature is maintained in the 500 to 600° F range for best soldering effect.

Before You Solder It In

A useful thing to do before you solder in a replacement part is to test the device in the circuit. Simply insert the chip or other device into the solder holes and wedge each lead in its hole with a toothpick. Then energize the circuit and test. After proper function is assured, remove the toothpicks and solder the component into the board.

CIRCUIT BOARD REPAIR

Repairing damaged circuit boards is a lucrative business, and several companies have developed around this activity. For some board failures, you can repair your own circuitry and save time and money. As a technician, you may be required to do circuit board repair as part of your job.

Before soldering in new components, check over the board for any broken traces or pads lifting off the board. If a trace is open and is starting to lift away from the board surface, jumper across the broken spot from one component solder pad to another pad. Use solid No. 18 or No. 20 wire tinned at both ends before soldering.

If a pad or trace lifts free, replace it with an adhesive-backed pad or trace overlapping the damaged area. Scrape the coating off the pad or both ends of the trace so the new pad or trace can be soldered firmly to the existing pad or trace. Remove all excess solder and redrill any lead hole that has become covered or plugged with residual solder.

RECOMMENDED TROUBLE-SHOOTING AND REPAIR EQUIPMENT

As a serious hobbyist or technician you'll be tackling some tough problems. You can minimize your investment costs and yet optimize your chance of success by carefully selecting your equipment and tools.

First, make sure you have a set of good screwdrivers–both flathead and Phillips head. Maintain a wide selection of sizes–from the tiny "tweakers" to an 8-inch flat head. You might also find a set of jeweler's screwdrivers helpful.

Maintain several sizes of long-nose or needle-nose pliers, and several sizes of diagonal cutters or "dykes" for cutting wire and pins. A good low wattage soldering iron whose tip temperature is automatically controlled is a must if you intend to replace nonsocketed components. A simple $3^1/_5$ digit DVM or DMM is useful for test measurements. Another handy tool is the logic probe.

You'll need a 15 to 25 MHz oscilloscope with dual trace and a time-base range of 200 nanoseconds to 0.5 second. Select a scope with a vertical sensitivity of 10 millivolts per division or better.

You can get by quite nicely using the probe, pulser, tracer, and DMM as your primary equipment. An oscilloscope will make certain failures easier and faster to find.

SPARE PARTS

Finding that a trouble exists is only part of the problem. You must locate the specific chip (if the software doesn't) and then make the repair. This, too, can be somewhat challenging. Fortunately, most of the chips on your IBM PC system board are standard 74xx and are readily available. Just about any electronic parts store will have a supply of 7400 series chips. Most of the 8xxx series chips are also easy to obtain. You can make a list of the spare parts you'd like to have on hand from the listing in Appendix A.

Because of the cost involved you will probably want to maintain a minimal stock of repair parts; yet you want to be able to fix a machine quickly when it breaks down.

The optimal backup would include one each of every type of chip on the IBM PC's system board and adapter cards. This represents an investment of $100 to $200 in 150 or more chips. Now, IBM PC custom chips–the ROMs–are available from authorized IBM service centers. The total number of chips in the PC is higher than the number of chips you need as spares because many of the same type chips are used in different places on the system board. In addition, you only need a few of the RAM chips as spares. Your largest expense in chips will be for the ROMs (unless you are using an 8087 coprocessor chip).

Several companies are marketing spare parts packages with schematics, diagnostic tests, and one each of the chips for the IBM PC.

The custom IBM PC chips–the five ROMs –are proprietary to IBM, and I recommend you buy all these custom chips directly from IBM or from your local IBM PC repair center. Most chips for the IBM PC motherboard are generally easy to locate and are also inexpensive.

Should you be using an 8087 coprocessor in your PC, and need to replace the 8087 or the 8088, be aware that these are often sold as a matched pair. To see if your 8088 will work properly with your 8087, check the copyright date marking on the top of the chip. If the copyright date is 78, it *may* work with the 8087. If the date is '78 '81, it is certain to work with the coprocessor chip. Any other date stamp might not work unless it is stamped P8088 or D8088 followed with S4716. These revisions of earlier 8088 devices will also work.

Recent articles in *PC Magazine* suggest that the NEC V20 processor is a direct replacement for the 8088 CPU while providing a 5 to 10 percent processor speed improvement. The NEC V20 is also compatible with the Intel 8087 match coprocessor.

The trade magazines sometimes advertise inexpensive packages of IBM PC repair parts. Some of the PC chips can be replaced only by a board exchange. The exchange price may seem high, but you do get a new board, and the type of failure that requires this action does not occur very often. In fact, it occurs very infrequently.

In Appendix A you'll find a listing of each chip in your computer including its designation, name, and location.

SUMMARY

Customers expect you to be perfect. With the computer field changing rapidly, it's tough to keep on top of new developments. You need books like this technical manual to increase your knowledge, learn new techniques, and avoid the mistakes of poor troubleshooting–such as using a bare cotton swab with low-grade alcohol, "cleaning" a disk drive read head, wiping a soldering iron on a wet sponge just before putting it in its holder. These are common errors of poorly trained (or poorly motivated) technicians. After reading this book, you know the right tools to use and the right procedures to follow to troubleshoot and repair failed systems in minimum time.

4

Preliminary Service Checks

Conducting preliminary service checks before actually digging into a problem can quickly find minor malfunctions. Most of the time all that you find is a basic problem. Even the most common system problem (disk drive tracking and speed) can be solved by preliminary checks. But there are many other checks that can be made to isolate a failure. Disassembly and reassembly instructions can be found in the appendix.

The first (and probably most important) step to take when troubleshooting a problem is to determine what conditions start (or stop) the failure. A defect can often be started by some action on your part. This helps to localize and isolate the problem much faster. It also beats trying to find problems that start and stop on their own, totally ignoring actions by you. Once this is accomplished, begin conducting preliminary service checks by disconnecting all unneeded external peripherals. This includes display units, printers, joysticks, modems, and mouse tablets. Connect your own service monitor that has been previously verified fully operational. Then check all internal and external interface cables. Look for corrosion or broken pins. Clean all edge connectors on the cards plugged into the expansion slots and disk drives. You're trying to remove these components as problem sources.

Not all the preliminary checks that will be covered in this chapter are necessary in every case. For example, if you turn on the computer and it runs through self test and boots, but no screen display appears, you don't want to waste valuable time verifying the output of the power supply, but you might want to check that the power is present on the video interface card. Don't get snookered into rotating adjustment controls to see if they have any affect on the symptom. Excessive rotation can obscure the symptom you're investigating and possibly cause other nonrelated problems. Even when the point is reached where the control seems to be in need of adjustment, do so carefully with minimum rotation. You don't want the drawbacks to outweigh the advantages. Follow a logical procedure in troubleshooting and you'll seldom go wrong. In fact, you'll find the problem faster and with far fewer headaches. In the following paragraphs, the recommended service checks have been arranged by specific failure symptom.

NO POWER

Turn the PC power on, and check for any indication of power (display screen brightens, beep sound, disk drive light comes on, and so forth). Turn off power to the computer. Unplug the power cables P8, P9, P10, and P11.

Reapply power and check plug P9 for +5 volts on pins 4, 5, and 6, and -5 volts on pin 3. Check plug P8 for +12 volts on pin 3, and -12 volts on pin 4. Check plugs P10 and P11 for +12 volts on pin 1 and +5 volts on pin 4.

If all voltages are missing, check for an open fuse F1 (Fig. 4-1). If F1 is open, replace the fuse, retest, and return the system to service. If F1 is not open, go to Chapter 5, section "Won't Boot, No Fan, Screen Blank."

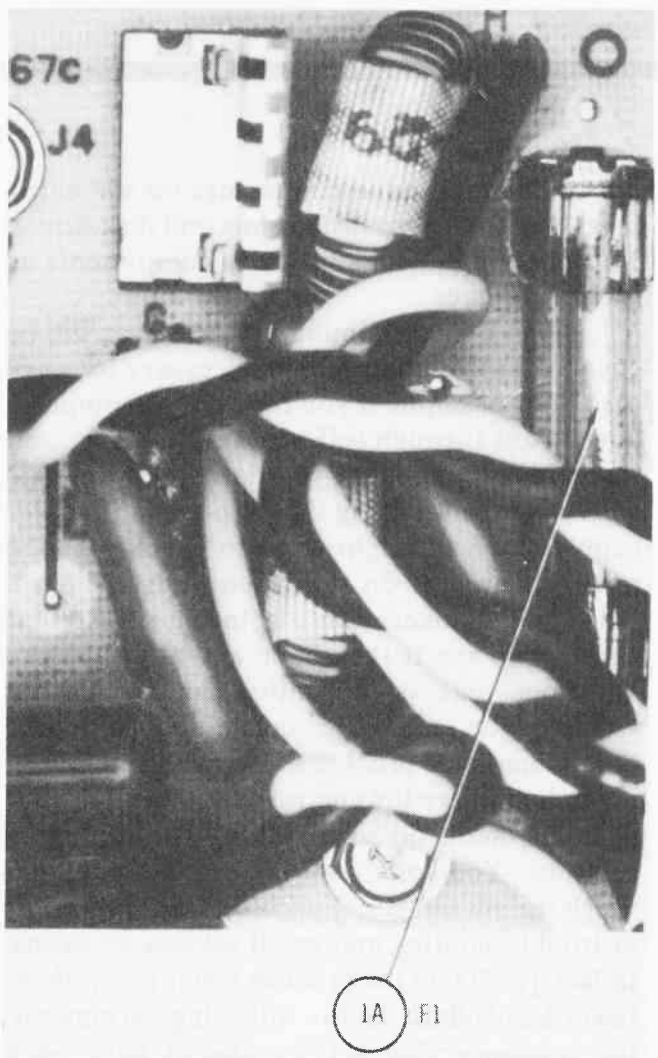


Fig. 4-1. The fuse is plainly visible on the far right.

If one or more voltages are missing, refer to the appropriate power missing symptom section in Chapter 5.

If all the voltages tested good at the plug pins, turn off power and reconnect the power plugs to the system board. Turn power back on and recheck for proper voltages on the same plug pins tested earlier. If the voltages are still present, turn off the power and reinstall one of the peripheral cards. Turn power back on and retest the plug pins for the continued presence of proper voltages. If the test passes, power down and reinstall another peripheral. Power up and retest for the presence of all proper voltages. If the test passes, continue with another peripheral. At some point you will discover that some or all the voltages are lost after reconnecting a peripheral. The last peripheral installed when the power failure occurred has experienced a component failure.

Look for anything that could cause a short on that peripheral's adapter card. Troubleshoot the adapter board and peripheral. See Chapter 5 for analyzing the monochrome display/printer adapter, the color/graphics adapter, and the floppy disk drive adapter cards.

SYSTEM BOARD PROBLEM

System board failures can be caused by the loss of the +5 volt power or by the loss of clock pulses. Therefore, test the clock circuit immediately after power up tests. An absent, jittery, or noisy clock signal can adversely impact the operation of the rest of the system. A bad clock signal can prove fatal to circuit operation or cause marginal operation leading to transient problems that are difficult to find. The best test sequence is to check power, clock, ROM, RAM, the I/O ports, and interrupts and bus control logic. You could divide the system board into two sections containing the CPU circuitry (8088, 8087, clock generator, bus controller, and buffers) and another section containing the standard peripheral I/O devices, RAM, and any analog circuits.

If power loss has not occurred, most start-up failures will result in a system lock with an error code displayed on the screen. If an error code appears, refer to the next section, "Self Test Error Code Displayed." If no error code appears, check for +5 volts on pin 18 of Clock Generator IC U11. Check for +5 volts on pins 31 and 40 of the 8088 CPU U3. Then check for a 14.31818 MHz timing signal on pins 16 and 17 of U11.

Check for 4.77272 MHz on pin 8 of U11 and on pin 19 of U3. Check that the switch settings on the system board are set correctly. These switches are tested by the system board circuitry to identify the hardware configuration and the amount of memory installed. Table 4-1 shows the switch settings.

If the system didn't boot, refer to the Won't Boot, Fan Works, Screen Blank section in Chapter 5. If boot-up occurs, monitor the self-test for audio or visual indications of system problems. Most system board failures can be captured by the self-test error routines located in the bootstrap ROM and identified by a speaker beep code and/or visual error code display.

Table 4-1. Switch Settings

Switch Block 1	Description
1, 7, 8	5¼ Disk drive setting 1 on, 7 on, 8 on = No drives 1 off, 7 on, 8 on = 1 drive 1 off, 7 off, 8 on = 2 drives
2	Presence of coprocessor 2 on = no coprocessor installed 2 off = coprocessor installed
3, 4	Total memory setting 3 off, 4 off = 64k memory or more 3 on, 4 off = 48k memory 3 off, 4 on = 32k memory 3 on, 4 on = 16k memory
5, 6	Monitor type settings 5 on, 6 on = no monitor 5 off, 6 off = monochrome monitor/ printer adapter or more than one monitor 5 off, 6 on = 40 x 25 color monitor 5 on, 6 off = 80 x 25 color monitor

Switch Block 2	Description
1-8	Total memory size setting (works in conjunction with SW1 3,4) 1,2,3,4,5 on, 6,7,8 off = 64k or less 1,3,4,5 on, 2,6,7,8 off = 128k memory 1,2,4,5 on, 3,6,7,8 off = 192k memory 1,4,5 on, 2,3,6,7,8 off = 256k memory

SELF-TEST ERROR CODE DISPLAYED

Turn off the computer, wait 7 seconds and turn power back on. Observe and listen to the system self-test. If no problems occur, the PC speaker will beep once and try to boot the master disk drive. If no disk is installed in the drive (or no drive connected to the PC), the operating system will shift into ROM BASIC. However, any failure of the self-test will generate an error code message and possibly an accompanying audio beep pattern. The audio beep indicators and system error codes are listed in Table 4-2.

If the system self-test detects a RAM failure, a four-character error code followed by the number 201 will appear in the top-left corner of the screen. The value 201 identifies a RAM problem; the four-character code defines the bank and row of memory ICs in which the error occurred. The first two characters refer to the memory bank in which the failure occurred. The last two characters define the bit position of RAM failure in a particular bank of memory. Table 4-3 translates error codes to malfunctioning bank and bit position. For example, a failure in the bit 5 position of bank 1, a code 0420 201 should appear in the upper left corner of the screen.

If the third and fourth characters in the error code don't match the codes in Table 4-3, swap the entire bank of RAM chips and try again. Another technique is to power down and swap each chip in the bad bank one at a time with the same bit position chip in an adjacent bank. Then power up and retest. When the error code shifts to the adjacent bank, the last chip swapped, was bad.

On the 64K/256K boards, a four character code is also displayed when a ROM failure occurs. The self-test routine resides on ROM U33. This ROM does not get tested by the self-test program. Table 4-4 can be used to translate the

Table 4-2. Beep Indicators and Error Codes Associated with the IBM PC Self Diagnostic Program

Beep Indicators

Indicator	Failure Location
No beep, nothing happens	Power, power supply
Continuous beep	Power, power supply
Repeating short beep	System board
1 long, 1 short beep	System board
1 long, 2 short beeps	Display circuit
1 short beep, blank or incorrect display	Display
1 short beep, Cassette BASIC display, no disk boot	Diskette, disk drive

I/O Error Codes

Code	Problem
199	Printer adapter card or printer malfunction
432	Printer adapter card or printer malfunction
7xx	System unit I/O malfunction
9xx	System unit I/O (parallel printer adapter) malfunction
901	Printer adapter card or printer itself is bad
11xx	System unit malfunction
12xx	System unit malfunction
13xx	Game control adapter card malfunction
14xx	Printer interface malfunction
15xx	System unit or communications adapter cable malfunction
18xx	Expansion unit or cable malfunction
1819	Expansion unit malfunction
1820	Expansion unit cable malfunction
1821	Expansion unit cable malfunction
20xx	System unit or communications adapter cable malfunction
21xx	System unit or communications adapter cable malfunction

Other Error Displays

Display	Meaning
Blank display, beep, drive starts to boot, but no Cassette BASIC message on screen	System Monitor BIOS ROM (U33) 8284 clock generator bad
KEYBOARD NOT FUNCTIONAL	Keyboard problem
PARITY CHECK 1	Power supply problem
PARITY ERROR 1	Try reseating RAM chips
PRINTER PROBLEMS	Printer problem, check interface

System Error Codes

These error codes can appear alone or in conjunction with other numbers.

Code	Problem
02X	Power supply problem
100	Option configuration wrong
199 100	Software option configuration installation wrong. Check switches.
101	System board malfunction
131	Cassette port error
201	RAM failure
xxxx = 201	Memory failure
1055 = 201	DIP switches set wrong
2055 = 201	DIP switches set wrong
xxxx = 201	RAM chip malfunction
PARITY CHECK x	
301	Keyboard malfunction, keyboard cable disconnected
xx301	Keyboard circuitry malfunction (xx is a hexadecimal value representing the scan code of the malfunctioning key)
401	Monochrome adapter card malfunction
501	Color/graphics adapter card malfunction
601	Diskette or disk drive interface malfunction (drive adapter, cable, drive A)
606	Drive assembly or drive adapter malfunction
607	Disk is write protected; disk not inserted right; write-protect switch bad, analog card malfunction
608	Diskette is bad
611	Drive data cable or disk drive adapter card is bad
612	Drive data cable or disk drive adapter card is bad
613	Drive data cable or disk drive adapter card is bad
621-626	Drive assembly is bad

four character error code to a specific ROM failure.

To check the ROM, substitute the suspected bad ROM IC with a known good ROM and retest. If the problem remains, refer to Chapter 5.

Table 4-3. RAM Memory Failure Error Codes

XX System Board Memory (Bank xx)	*XX 201 Parity Check 1* Failed Chip
00 = Bank 0	00 = Parity
04 = Bank 1	01 = D0 chip
08 = Bank 2	02 = D1 chip
0C = Bank 3	04 = D2 chip
	08 = D3 chip
	10 = D4 chip
	20 = D5 chip
	40 = D6 chip
	80 = D7 chip

Table 4-4. ROM Error Codes

Display	Problem
F600 ROM	Cassette BASIC ROM (U29) bad
F800 ROM	Cassette BASIC ROM (U30) bad
FA00 ROM	Cassette BASIC ROM (U31) bad
FC00 ROM	Cassette BASIC ROM (U32) bad

DISPLAY PROBLEMS

During system power-up, one of the BIOS routines initializes and starts the 6845 CRT controller and tests the video read/write storage. The program causes the 8088 CPU to check the setting of the video switch SW1 by reading port A of the 8255 PPI. This port is connected to the two configuration switches SW1 and SW2. The CPU logically ANDs the port data with the hex value 30H thus checking the settings of switch SW1-5 and SW1-6. If they are not off, the program jumps to a subroutine that tests to see which type video card is installed. If SW1-5, 6 are off, the program goes into an I/O memory parity test and then into setting the video mode. If a parity error occurs, a message is displayed. If a problem occurs setting the video mode, another error message is displayed.

The program also conducts a test of the video storage memory. If a failure occurs, the speaker is beeped. By reading the CRT controller status port and logically ANDing the reading with binary 1000, a test is made to see if the video/horizontal line changes state. If it does not go low during this timed test, a timer clocks out and an error message is displayed and the speaker is caused to beep. Several times during this power-up testing, the INT 10H video I/O procedure is called.

Color Display Problems

No Video

The first thing to do is to localize the failure to the display unit or the PC system unit. Do this by connecting your own test monitor with known good cable to the PC and retest. If the problem is corrected, try the original display unit with your good video cable. If it works fine, replace the video cable and return the failed unit to service. If it didn't work, the display unit has a malfunction.

If no video is displayed using your test monitor and cable and no cursor appears, check the configuration switches on the system board, particularly SW1-5 and SW1-6 which configure the system for the particular type monitor adapter board you are using. Clean and examine the pins of the edge connector of the adapter card. On the older adapter cards, the bracket from the adapter must be connected directly to the chassis to provide a suitable system ground for proper operation.

If the display malfunction persists, check for proper +5 volt power on the adapter board. If the voltage is improper, refer to the power supply problems in Chapter 5. If +5 volts is present, refer to Chapter 5 "No video." If the system

seems to work fine without the color card installed, but shuts down when it's mounted in the expansion slot, suspect ICs U26, U42, U60, U66, or U67.

Horizontal or Vertical Sync Problem

If no horizontal or vertical sync is working, suspect ICs U21, U63, U67, and U101. Refer to Chapter 5.

Cursor Problem

If the cursor is not blinking or is missing, suspect U12. Go to Chapter 5.

Fading or Wrong Color

Fading color or the wrong color can be caused by U20, U22, U43, U44, U45, U65, or U67 malfunctioning. Go to Chapter 5.

Video Memory Problem

Video RAM problems can be caused by failure in ICs U50 through U60. The error code 501 defines a malfunction in the color/graphics adapter card. Go to Chapter 5.

Monochrome Display Problems

When failure symptoms occur on systems that have the monochrome display/printer adapter installed, follow these steps when conducting preliminary checks. Connect the system to your test monitor with your own video cable to eliminate the display unit and cable as the problem cause. Did one long and two short beeps occur during power up? Has a 401 error code appeared on the display?

No Video

Check for +5 volts on the adapter board. If the voltage is present, refer to the monochrome video section of Chapter 5. If +5 volts is not present on the adapter board, troubleshoot from the +5 volt power input to the point of loss and correct as appropriate.

Cursor Problem

If there is no cursor on the screen of a system that was just brought to you, check the system board configuration switches SW1 and SW2. Pay particular attention that SW1-5 and SW1-6 are both in the OFF position and have not malfunctioned. Clean, and reseat the monochrome adapter card into the expansion slot.

If the cursor is not blinking or is missing, suspect U55 (DM74LS174N). Go to Chapter 5.

System Shuts Down With Monochrome Card Installed

If the system shuts down, one of the following ICs on the monochrome monitor/printer adapter could be bad. Refer to Chapter 5.

U3–DM74LS08N
U35–MC6845P
U45–74LS74APC
U54–DM74S86N
U64–DM74LS244N
U100–74LS32N
U101–74LS74PC

Printer Problems

As a group, printer problems rank somewhere between a pain in the neck and a minor nuisance depending on how much you depend on printed output. At the least, a malfunctioning printer means you'll have to MODEM over to another machine with a printer, transfer disks to another machine with a printer, or settle for display output only. For a writer like myself, loss of a printer can be catastrophic.

When printing ceases (or never begins), like the video output, you have three possible sources for the problem: the system board, the adapter card, or the printer itself. One of the few preliminary tests that you can accomplish, is conduct a printer self-test to see if the printer itself is functioning properly. Power down the

IBM PC itself. Keep the printer powered-up and, following the printer operating manual, conduct a self-test operation. When you are satisfied the printer itself is good, turn off power to the printer and check the printer cable interface to ensure no pins are corroded or have somehow broken or have become bent. The fastest way to locate a printer problem is to carefully evaluate the configuration and decide if any changes have occurred. Did you recently disconnect the printer? Did you recently replace the ribbon or paper in the printer? Perhaps a limit switch got bent or is sticking. Error codes 199, 432, or 9xx suggest a failure in the printer adapter card or the printer itself. An error code of 14xx defines a failure in the printer interface itself.

If you have a spare, connecting another printer and Centronics-parallel cable to the system unit is a good way to localize the problem to the printer adapter card.

If you suspect a system board failure, check the operation of the other peripherals–the video display and the disk drive interface–because these devices use the same system board signals during operation.

If you've localized the problem to the printer adapter card (or the monochrome monitor/ printer adapter if installed), refer to Chapter 5 for detailed circuit troubleshooting.

System Shuts Down With Printer Card Installed

If the system works fine as long as you don't plug the printer adapter card into an expansion slot, but shuts down whenever you try to boot up with the card installed, suspect U1 data transceiver 74LS245, U9 2-input NOR gate 74LS02, or U11 2-input XOR 74LS86.

A total system shutdown with the monochrome monitor/printer adapter installed can be caused by the failure of U3 2-input AND 74LS08, U35 MC6845 CRT controller (its on the same board with the printer circuitry), U45 D flip-flop 74LS74, U54 2-input XOR 74LS86, U64 tristate octal buffer 74LS244, U100 2-input OR 74LS32, or U101 D flip-flop 74LS74. Go to Chapter 5.

Printer Won't Print

If you are using the standard printer adapter card and the printer just won't print a thing (after successfully conducting a printer self-test) suspect U1 data transceiver 74LS245, U2 line driver 74LS240, U6 1-of-4 decoder 74LS155, U7 flip-flop 74LS174, or U8 open collector hex inverter 7405 (see COMPUTERFACTS pages 7 and 54).

If you have the monochrome monitor/ printer adapter card installed and the printer won't print, suspect U37 tristate buffer 74LS240, U38 open collector hex inverter 7405, U39 D flip-flop 74LS174, U56 hex inverter 74LS04, U57 2-input NOR 74LS02, or U61 1-of-4 decoder 74LS155. Go to Chapter 5.

Prints Garbage

If you are using the printer adapter card and the printer starts printing random, or garbage, characters, suspect U3 data buffer 74LS244, or U4 output latch 74LS374. Refer to Chapter 5 for all circuit board troubleshooting and repair.

Keyboard Problems

This is a tough problem to have because it's your primary means for communicating with the PC.

No Keys Respond

If depressing the keys causes no response, disconnect the keyboard from the system unit and check the keyboard cable for continuity. Look for corrosion or broken pins. Do a continuity test for open wires in the cable. If this is not the problem, disassemble the keyboard, reconnect it to the system board, energize the system and check to +5 volts on pins 26 and 40 of 8048 single chip computer M1 (labelled 8340X7).

If the failure occurs during power-up, the error codes are helpful in localizing the failure. As shown in Table 4-1, keyboard error code 301 describes a keyboard or keyboard cable failure. A xx301 error code defines a keyboard circuitry

failure (xx is hexadecimal value representing scan code of failed key). Go to Chapter 5.

Single Key Won't Work

If a single key malfunctions (and your keyboard keys can be removed), replace the bad key. Otherwise, replace the keyboard.

Get Unwanted Repeat Key Action

If a key is pressed and an unwanted repeat occurs, refer to Chapter 5 for circuit troubleshooting. If many different characters are printed when only one key is depressed, the keyboard is bad or the 8048 inside just blew its silicon and should be replaced.

Disk Drive Problems

When a disk drive fails to boot a system disk, one of several things could have failed, the disk itself is bad, the drive analog card failed, the drive mechanics are out of tolerance, the disk drive adapter card in the expansion slot is bad, or the system board itself has a failure. It could just be that you inadvertently placed configuration switch SW1 positions 1, 7, and 8 ON showing that you don't have *any* $5^1/_4$ inch drives attached.

The boot-up diagnostics are handy here. If you get a short beep, the "Cassette BASIC" display, and no disk boot, the problem is probably the diskette or in the drive itself. An error display of 60X (501, 606, 607, 608, 611, 612, 613, or 621 through 626) indicates that a malfunction has occurred in the disk system (diskette, disk drive, disk drive interface, or the adapter board). Refer to Table 4-2 for the meanings of the 60X error codes.

When a disk-related malfunction occurs during operation, you must verify that the problem is not in the diskette being used. Reboot with a good copy of the system disk. If the disk checks good, try to reboot with the same disk that you were using when the problem occurred. If a failure occurs, the disk is bad (or needs reformatting).

No Disk Boots in Drive A

If no disk will boot-up and run in drive A, power down, unplug the drive data cable from drive A and connect it to drive B–also, swap the jumper at 1E on the analog card between drives A and B. Power up again and reboot using drive B. If drive B boots normally, drive A has a malfunction. If not, power down and check the cable for continuity. Clean and examine the cable and the adapter card edge connectors. Check for +5 volts on pin 4, and +12 volts on pin 1 of the power supply cable. If the power at the plug is normal, troubleshoot the disk drive adapter card.

Both Drive Lights Come On

If both drive run lights come on when reading or writing a disk, check the cables to see if you misconnected the drive A data cable to drive B. An active low signal on any of the MOTOR ON, MOTOR ENABLE (A), DRIVE SELECT (A), or DRIVE SELECT (B) lines can cause the led CR27 to energize (see disk drive CF page 2). Another possible cause could be failure of U16 output NAND 7438 or U17 output D flip-flop 74LS273 on the adapter card.

Seek Error Message

If you get a SEEK ERROR message displayed, a track is unreadable or head misalignment has occurred. Try to read another disk. If this doesn't work, suspect U4 2-input AND gate 74LS08, U6 floppy disk controller D765AC (UPD765), or U18 tristate octal inverter buffer 74LS240. Refer to Chapter 5.

Drive Destroys Write-Protected Data

If a drive destroys data on a write-protected disk, suspect a failure in U6 FDC UPD765, U10 multiplexer 74LS153, U11 D flip-flop 74LS175, or U18 tristate octal inverter buffer 74LS240. Refer to Chapter 5.

Can't Read From Either Drive

If you can't read data from either drive, suspect U6 FDC UPD765, U7 driver MC3487, U9 open

collector 2-input NAND 7438, U18 buffer 74LS240, U22 J-K flip-flop 74LS112, U23 4-Bit counter 74LS112, U25, another 4-bit 74LS112 counter, or U26 2-input NOR 74LS02.

Drive Light On But No Data To Memory

If a disk drive run light is on, but data is not going into memory check that the drive motor is actually running. Another cause could be the motor speed. If it drifts out of tolerance, reset it using speed adjust control (R4A). This will be covered shortly. It's also possible that the motor drive belt is loose. Check the belt and replace it if it appears loose.

Can't Read or Boot DOS Diskette

Switch drives A and B data cables–also, swap the jumper at 1E on the analog card between drives A and B. Reboot the DOS diskette in drive B. If it works, drive A needs track 0 alignment. If the disk won't boot in drive B either, try another DOS disk. Suspect that the original DOS disk lost part of its format or directory data.

Drive Operates Intermittently

This problem can be caused by dirty, clogged write/read heads, speed out of tolerance, or mechanical alignment of the drive mechanics. Conducting periodic head cleaning (after every 40 hours of operation) will help keep debris on the head from destroying data stored on the disk. Periodic maintenance on the disk drive speed will eliminate this as a possible intermittent failure cause.

Drive Writes/Erases Data On Write-Protected Disk

Switch active drives and try another write-protected disk that contains nonimportant information. Try to write to this disk. If the drive writes, suspect a failure on the adapter card. If the drive does not write, a failure has occurred on the original disk drive analog board. Refer to Chapter 5.

Disk Drive Head Cleaning

Heads need cleaning to remove the oxides from the disks building up on the leading edge of the head (the side facing the direction of disk rotation), as shown in Fig. 4-2.

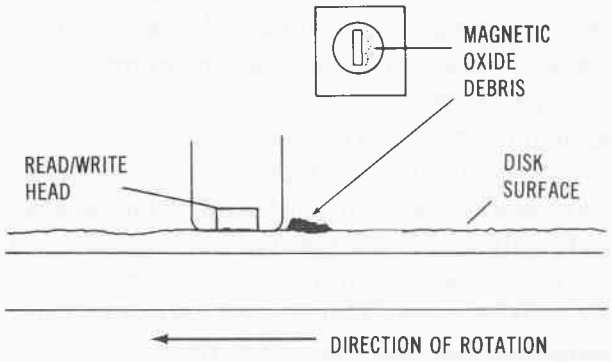


Fig. 4-2. Oxides are wiped off the disk surface and build up on the head surface.

Head cleaning diskettes of various kinds are available. The "wet" diskette kind works with a cleaning solvent. Some head cleaners are abrasive and can damage the head if they are used for too long. With this type of cleaner, you must use the cleaner just long enough to remove the oxide build-up but not long enough to damage the head.

Nonabrasive head cleaners are also being marketed. Two examples are Verbatim's Datalife head-cleaning kit, and Innovative Computer Products' Perfect Data head-cleaning kit. Both products use fabric-covered disks which are dampened with a cleaning solvent. With the one kit, you sprinkle cleaning solvent on the disk fabric and then insert the disk into the drive for spinning action head cleaning. The disk can be used as many as 13 times. The other kit has cleaning disks that are predampened and individually sealed. A cleaning disk is used once and then thrown away, using another the next time. Both of these products work well.

Since any cleaning disk works by rubbing action and chemical action between the disk fabric and the drive head, there is potential for abrasion to occur. So you must be careful not to leave the disk spinning in the drive for too long. A cleaning disk can be allowed to spin in a disk

drive for 30 seconds with no apparent damage. With most cleaning disk kits, 45 seconds is too long to keep the cleaning solvent in contact with the drive head.

Drive heads can also be cleaned with alcohol and a cotton swab wrapped in a lint-free material (see Fig. 4-3). With manual alcohol and swab cleaning, you could accidentally scrub the pressure pads by mistake, causing more problems than you're preventing. But, if you're careful, manual cleaning can be effective.

Special cleaning material such as cellular-foam swabs and chamois leather cloth are good materials to use for manual head cleaning. Or you can use a piece of bed sheet wrapped around the cotton swab. Uncovered cotton swabs are dangerous because the cotton fibers can catch or pull away and lie in the drive or on the head,

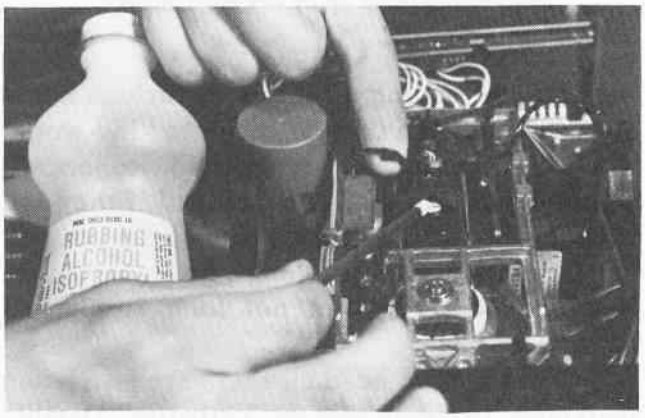


Fig. 4-3. Drive read heads can be cleaned using denatured alcohol and a lint-free swab.

becoming cotton logs on a disk-surface highway, waiting to get swept into the drive head that rides on the surface of the disk. These fibers can also catch on the ferrite chip in the middle of the ceramic head, loosening it from its mounting and ruining the head. Surgical isopropyl alcohol or methanol can be used as the cleaning solvent. The solvent used must not leave a residue when it evaporates, so most other alcohol solvents should be avoided. You can also use typewriter cleaner or trichloroethane. Always use plenty of ventilation and make sure the solvent has evaporated before you operate the drive.

How often the head must be cleaned depends upon how much the drive is used and what type of diskettes are used. A quality diskette is good for about 3 million passes, or rotations, against a read/write head before enough oxide is worn off so that the head needs cleaning. The "bargain" disks are good for one-tenth the rotational life. This means that instead of 167 hours of access time, you might get 16 hours before the head gets caked with oxide or a disk surface gets too worn to write to or read from. This is why bargain disks don't seem to last very long.

A useful rule of thumb for head cleaning is to clean the read/write head every 40 hours of disk operation. This means clean after 40 hours of rotational life if standard disks are being used. A popular approach is to wait until read/write errors start to occur, then replace or clean the head.

Keeping the disk drive door closed unless inserting or removing a disk will help keep dust and dirt out. It also prevents unwelcome visitors (insects and even mice) from climbing into the drive.

To Clean the Drive Head Using a Cleaning Disk:

1. Turn computer power on.
2. Dampen the cleaning disk with the solvent supplied.
3. Insert the dampened cleaning disk in the drive.
4. Close the drive door.
5. If in BASIC, reset the system (warm boot). With the cleaning disk inside the drive, the disk will simply spin, cleaning as it whirs along.
6. After 20 or 30 seconds, open the drive door and remove the disk.
7. Turn off the computer.
8. Let the drive read/write head dry thoroughly before operating the system.

To Clean the Drive Head Manually:

Tools Required:

Flathead screwdriver

Phillips head screwdriver
Protective pad
Adequate lighting
Tray to hold loose screws

1. Turn off power to the computer.
2. Disassemble the computer using the procedures found in Appendix D.
3. Disconnect the disk drive data cable from the back of the drive.
4. Remove the two silver flathead screws holding the drive tight to the chassis, as shown in Fig. 4-4.

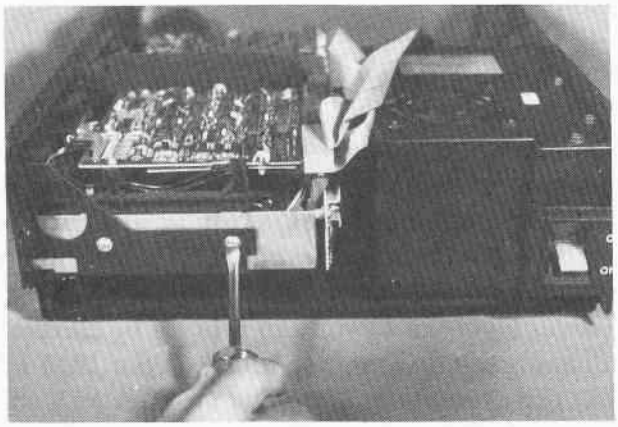


Fig. 4-4. Remove the two flathead screws holding the disk drive to the chassis.

5. Gently pull the drive forward about 2 inches from the front of the chassis.
6. Disconnect the power supply cable from the back of the analog card.
7. Gently remove the drive from the chassis.
8. Carefully remove the head cable(s) from connector (located in the front-right corner).
9. Carefully remove the two Phillips screws holding the analog card on the drive (located in the front on both sides), as shown in Fig. 4-5.
10. Slide the analog card toward the back of the drive until the card is free of the grooves in the drive mechanism. (You may have to wiggle the card gently to get it to slide.)
11. Lift up the analog card from the front.
12. Carefully lift the black head load arm as shown in Fig. 4-6, and look for discoloration (build-up) on the surface of the pad, or on the read/write head below.

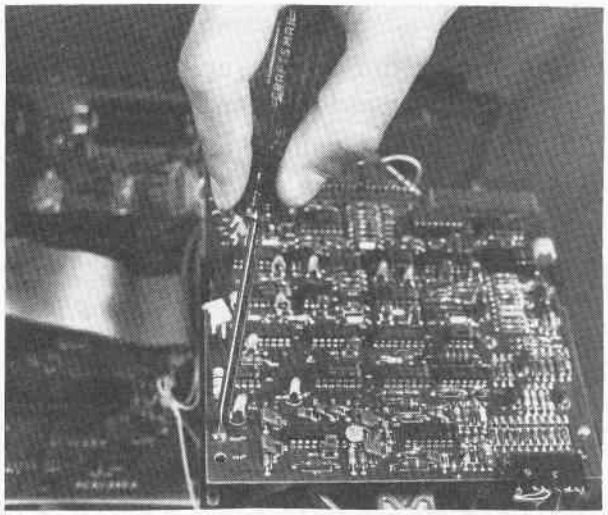


Fig. 4-5. Remove the two Phillips screws holding the analog card on the drive mechanism.

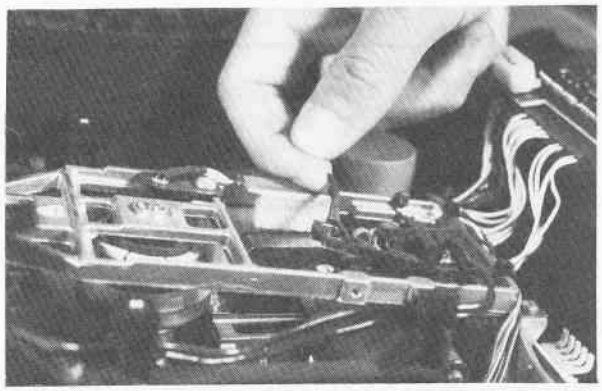


Fig. 4-6. Carefully lift the head load arm to expose the read head below.

13. Using a special foam or wrapped cotton swab dampened with cleaning solvent, gently rub the head and the pad (see Fig. 4-6).
14. Let the surfaces dry completely before reassembling.
15. When the head and pressure pad are dry, carefully slide the analog card back into the grooves in the drive mechanism.
16. Reinstall the two Phillips screws into the analog card (in the front on both sides).

17. Reconnect the head cable(s), being very careful not to break or short connector pins.
18. Carefully push the drive mechanism back into the chassis until about 2 inches of the drive housing is still exposed.
19. Connect the power supply cable to the back underside of the analog card.
20. Push the drive all the way into the chassis housing.
21. Replace the two silver flathead screws on the sides of the drive (refer to Fig. 4-4).
22. Reconnect the ribbon cable to the drive.
23. Power up the computer.
24. When all the surfaces are dry, place a copy of a program disk in the cleaned drive.
25. Close the drive door.
26. Load and run a program.
27. Reassemble the computer.
28. Restore the system to full operation.

> Note: if you have any problems refer to the beginning of the disk preliminary check section.

Disk Drive Head Cleaning Interval

Cleaning a drive head is like changing the oil in a car. It's usually done when you feel you've driven enough miles or when the oil looks dirty. Some software manufacturers recommend cleaning heads every other week. Some repair technicians say clean every six months. Others suggest you don't clean your heads until the disk drive makes mistakes trying to read or write data. Since no hard and fast rule has been offered, refer to Table 4-5 for some recommended head cleaning intervals.

If the drive is used in an area that gets a lot of smog, you may want to clean the heads more often. In any case, it won't hurt to clean at least annually. If read/write errors begin to occur, check the operational time log (if one is being maintained) to see if PMs are due. If the customer doesn't keep an operational time log, recommend that one be started. Some recommended operational log sheets are included in Appendix I.

Table 4-5. Disk Drive Head Cleaning Interval

System Usage	Cleaning Interval
Over six hours each day	Weekly
Daily	Monthly
Light to moderate	Every other month
Occasionally	Every six months

Disk Speed Tests

Variation in speed is caused by normal mechanical drive wear or by excessive moving and reconnecting drives. Just as automobile engines need periodic checkup and engine retuning, disk drives benefit from the TLC (tender loving care) you can provide by correctly adjusting the drive motor speed. Your IBM PC disk drives rotate at 300 rpm and work with soft sectored disks where the computer software identifies the beginning and end of each of the sectors on each of the tracks. No timing holes are used as with hard sectored disks. This makes the speed of rotation critical to the accurate synchronization of the software with the signals stored on the disk. If the speed is off by only 10 rpm, the drive may not be able to correctly read the disk.

Should the speed be incorrect, the data will be written in the wrong location on the disk. The next time an access is made to that area on the track again, the computer will hang up and generate a disk error. While disk speeds between 291 rpm and 309 rpm should be acceptable for read/write operation, speeds outside this range cause intermittent or disastrous results. If the speed becomes slower than 270 rpm or faster than 309 rpm, any write action will erase the synchronization timing marks on the disk making it useless unless the disk is reinitialized (wiping out the data that was already stored on the disk).

There are two ways to tune drive speed. You can adjust the speed using a disk speed test program, or you can adjust the speed using a standard room lamp. Both techniques require your removing the disk cover. A good way to conduct speed adjustments (and alignment of the drive mechanics) is to boot the adjustment/alignment program on a known good drive (like your test monitor, you keep one of these handy also) connected as drive A with the drive to be adjusted configured as drive B. Then you can boot the diagnostic test disk in drive A and control the out-of-tolerance drive even if it no longer will boot a disk itself.

Disk Drive Disassembly

Tools Required

Phillips head screwdriver
Flathead screwdriver
Jewelers flathead screwdriver
Protective pad
Tray to hold loose screws

1. Turn the power off to the computer.
2. Ground yourself to remove any electrostatic charge by touching the metal switch on a nearby lamp, or some other grounded surface.
3. Disassemble the computer as shown in Appendix D.
4. Disconnect the drive data cable from the rear of the drive.
5. Remove the two silver flathead screws holding the drive tight, as shown in Fig. 4-4.
6. Gently pull the drive forward about 2 inches out of the chassis.
7. Disconnect the power supply cable in the back of the analog card.
8. Pull the drive completely out of the chassis.
9. Set the drive mechanism on its side crossways on the top of the power supply.
10. Reconnect the drive cable and the power supply cable to the drive.

You are now ready to conduct drive speed adjustment.

METHOD 1—DISK SPEED PROGRAM

The test disk should have full instructions on how to perform the speed adjustment. The speed should be adjusted to 300 rpm ± 1.5 percent by rotating the 0 to 1000 ohm potentiometer R4A (R23 in some drives) on the analog card in the drive.

1. Locate the speed adjustment control potentiometer (R4A/R23) in the back middle of the drive, as shown in Fig. 4-7.

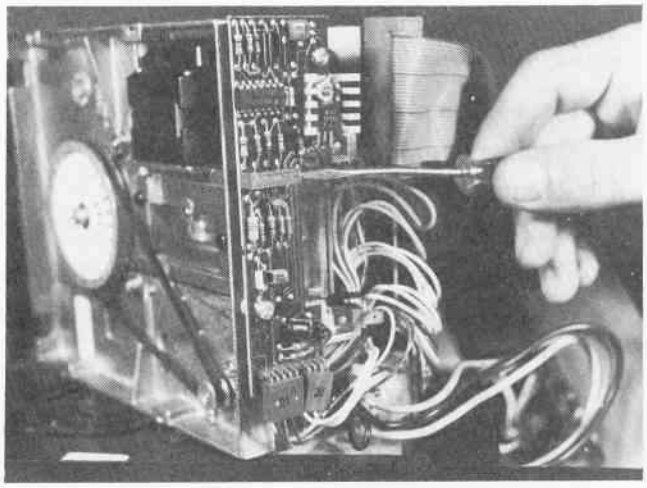


Fig. 4-7. The screwdriver is inserted in the speed adjustment potentiometer.

2. Reconnect the power cord to the computer.

> **Caution: Be careful not to touch inside the drive mechanism or the electronics with power on.**

3. Turn on power to the computer.
4. Insert the disk containing a disk speed test program in the drive to be adjusted.
5. Close the disk drive door.
6. Boot the disk in the drive.
7. Follow the test procedures.

Most speed test programs display a graduated scale of some sort as shown by the example in Fig. 4-8. Using a jewelers screwdriver or a "tweaker" (small screwdriver), slowly turn the speed control adjustment pot screw until the speed display shows the actual rotation time as close to 300 rpm as possible–within plus or minus 6 milliseconds.

```
                    DISK SPEED TEST

                          |
                          |
                          |
                          |
                  -10     |     +10
                   |      |      |
                   |      |      |
-100 >------------------- 0 -------------------< +100
                   | /\   |      |
                   | ||   |      |
                   | ||   |
                          |
                          |
                          |
                  CURRENT SPEED SETTING  =  -4

ESC(APE) TO STOP TEST
```

Fig. 4-8. Sample disk-speed test screen display.

8. Remove the disk.
9. Turn off the power to the IBM PC.
10. Disconnect the disk drive data cable and power supply cable from the drive mechanism.
11. Push the drive mechanism into the front of the chassis until about 2 inches of the mechanism is left exposed.
12. Reconnect the power supply cable to the back of the analog card.
13. Push the drive all the way into the chassis.
14. Reinstall the two flathead screws on the side of the drive.
15. Reconnect the disk drive data cable to the drive.
16. Turn on the computer and test operate the drive.
17. Restore the system to full operation.

METHOD 2—TUNING LAMP

The use of a tuning lamp enables speed adjustment without the need for a speed adjust program. This technique can be used if only one drive is available and it is the one in need of maintenance.

1. Disassemble the drive and set it on its side, as shown in Fig. 4-9.

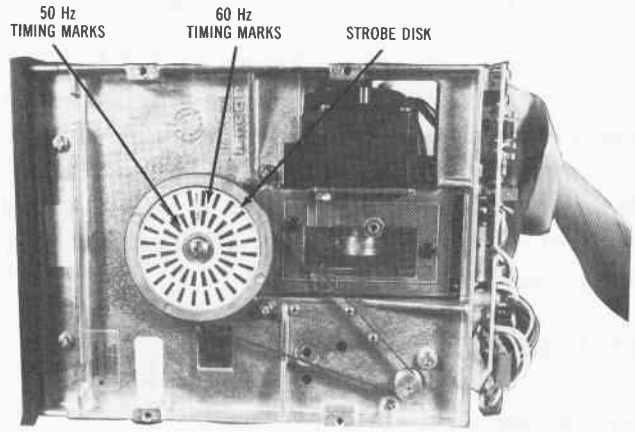


Fig. 4-9. The disk drive set on its side with the strobe wheel facing you.

2. Place a fluorescent light near the drive so the light from the lamp illuminates the speed strobe wheel on the bottom of the drive mechanism. An ordinary incandescent room lamp will work, but a fluorescent light is easier to use.
3. Locate the timing marks on the strobe disk, as shown in Fig. 4-9. The outer circle of markings is used for 60 Hz electrical systems such as are common in the United States. The inner circle of strobe markings is used with 50 Hz line power such as that found in Europe. This concept of speed adjustment is amazingly simple, and quite accurate. When you place the strobe disk in the light of fluorescent light and cause the disk to spin, you will notice that the strobe disk marks slowly rotate in one direction or another, depending on whether the disk speed is fast or slow. This action is much like the effect of movie film when stagecoach wheels seem to be ro-tating in the direction opposite the movement of the coach. The wheels are rotating at a different speed than the film is moving through the projector so you see this strange effect. The marks on the strobe

disk are spaced so they appear stationary when the speed is exactly 300 rpm.

4. With a lighted lamp near the drive, reconnect the disk data interface cable to the computer.
5. Reconnect the power supply cable to the drive in back of the analog card.
6. Power up the computer.
7. Insert a blank, nonformatted disk in the drive and close the drive door.
8. After the computer goes into BASIC, reset the system (warm boot).
9. Observe the strobe wheel as the disk spins in the drive.
10. Using a jeweler's screwdriver or a "technician's tweaker" (small screwdriver), adjust the speed control pot until the strobe disk seems to be sitting still.
11. Open the drive door and remove the disk.
12. Turn off the computer power.
13. Disconnect the disk drive data cable from the drive.
14. Disconnect the power supply cable from the back of the drive.
15. Gently push the drive mechanism into the front of the chassis until about 2 inches of the drive remains exposed.
16. Connect the power supply cable to the back of the drive analog card.
17. Gently push the drive mechanism all the way into the chassis.
18. Reinstall the two screws holding the drive into the chassis.
19. Reconnect the drive data cable to the IBM PC drive.
20. Turn on the computer and test operate the drive.
21. Restore the system to full operation.

DISK DRIVE ALIGNMENT

The alignment adjustments in the IBM drives set the positioning of the read/write head correctly over the tracks on the disk, adjust the disk stop guide, adjust the index hole sensor, or adjust the collet hub that fits in the hole in the disk. A check is also made of the head azimuth compared with the disk track. These procedures require special equipment, including a dual-trace oscilloscope, disk alignment tools, and special alignment disks. Use a Dysan Analog Alignment Diskette 224/2A containing alignment patterns and follow the procedures outlined in the following paragraphs (or described in the SAMS COMPUTERFACT CSCS2 page 43).

The most critical alignment adjustment is the read/write head alignment or tracking. Some programs require very accurate alignment of the head over the track. If the program loads fine but won't read data and the disk just spins, the drive may have tracking out of alignment.

If you have to replace the electronics analog card in the drive, readjust the tracking. Each card is tuned for the drive, and a new card could affect the head tracking.

Alignment should be checked every year. The easiest way to accomplish this is to format two disks on two different drives whose speeds have been verified to be correct. Save some programs on each disk using the same drive on which formatting was done. Read and write each disk with its drive to make sure the individual drives work satisfactorily. Then switch disks and see if each disk works properly in the alternate drive. If one drive reads correctly while the other can't find the data or reads out "garbage," you know you have alignment problems. These steps could be added to your maintenance program for each system you are responsible for.

TRACK 00 ADJUSTMENTS

Adjustment of the track 00 switch (SW2) properly positions the drive write/read head over the disk track 0. A stop screw holds the switch at a setting fixed at track 0. This switch setting rarely needs adjustment–if it does the drive will usually not read at all. If the system doesn't boot and you think that the track 0 alignment may be out of tolerance, then start by setting the radial

head alignment first. Configure a good known drive as drive A and the drive to be aligned as drive B.

To Adjust the Track 00 Switch (SW2)

Tools Required

Flathead screwdriver
Phillips head screwdriver
Protective pad
Adequate lighting
Tray to hold loose screws
Screw retaining cement

1. Turn off power to the computer.
2. Disassemble the computer using the procedures found in Appendix D.
3. Disconnect the disk drive data cable from the back of drive B.
4. Remove the two silver flathead screws holding drive B tight to the chassis, as shown in Fig. 4-4.
5. Gently pull drive B forward about 2 inches out from the front of the chassis.
6. Disconnect the power supply cable from the back of the analog card mounted in drive B.
7. Gently remove drive B from its chassis.
8. Set drive B on its side so that you can get to the adjustment screws at the back of the drive.
9. Connect the power supply cable to the back underside of the analog card.
10. Reconnect the drive data cable to drive B.
11. Insert a blank disk into drive B.
12. Insert the diagnostic disk in drive A and power up the computer.
13. Set the head on drive B to track 01. (Should be described on diagnostic program menu display.)
14. Loosen the retaining screw at the base of the track 0 switch bracket on drive B.
15. Turn the track 00 switch adjustment screw all the way counterclockwise.
16. Turn the screw back in a clockwise direction listening for the switch to click.
17. When the switch clicks, turn the screw clockwise one-half turn more and retighten the bracket screw.
18. Turn off power to the system.
19. Disconnect all cables to the drive.
20. Carefully push the drive mechanism back into the chassis until about 2 inches of the drive housing is still exposed.
21. Connect the power supply cable to the back underside of the analog card on drive B.
22. Push the drive all the way into the chassis housing.
23. Replace the two silver flathead screws on the sides of the drive, as shown in Fig. 4-4.
24. Reconnect the drive data cable to drive B.
25. Power up the computer.
26. Place a *copy* of a program disk in the drive, close the drive door and verify correct operation.

To Set the Track 00 Stop Adjustment Screw

Tools Required

2 channel scope
Flathead screwdriver
Phillips head screwdriver
Protective pad
Adequate lighting
Tray to hold loose screws
Test diskette
Alignment diskette

1. Turn off power to the computer.

   If the computer is not disassembled for maintenance, do the next step; otherwise, go to step 3.
2. Disassemble the computer using the procedures found in Appendix D.

3. Disconnect the disk drive cable from the back of drive B.
4. Remove the two silver flathead screws holding the drive tight to the chassis, as shown in Fig. 4-4.
5. Gently pull the drive forward about 2 inches out from the front of the chassis.
6. Disconnect the power supply cable from the back of the analog card.
7. Gently remove drive B from its chassis.
8. Set the drive on its side so that you can get to the adjustment screws at the back of the drive.
9. Connect the power supply cable to the back underside of the analog card.
10. Reconnect the drive data cable to drive B.
11. Connect scope channel A to TP1 on the analog card.
12. Set the sweep time to 10 microseconds and the voltage per division to 0.5 volt. Trigger on channel A.
13. Insert your alignment diskette in drive B.
14. Power up the computer.
15. Boot the diagnostic test disk in drive A.
16. Set the head on drive B to track 00. You should see a 0.6 volt, 125 kHz signal on the scope.
17. Turn the track 00 stop adjustment screw in a clockwise direction until the signal starts to decrease, then turn the screw one-half turn in a counterclockwise direction.
18. Cement the screw in this setting so it doesn't move during normal operating vibrations. (Some technicians use typewriter correction fluid, others use the green-colored cement to prevent the screw from vibrating out of its setting.)
19. Turn off power to the system.
20. Disconnect all cables to the drive.
21. Carefully push the drive mechanism back into the chassis until about 2 inches of the drive housing is still exposed.
22. Connect the power supply cable to the back underside of the analog card.
23. Push the drive all the way into the chassis housing.
24. Replace the two silver flathead screws on the sides of the drive, as shown in Fig. 4-4.
25. Reconnect the drive data cable to drive B.
26. Power-up the computer.
27. Place a *copy* of a program disk in the drive and operate the system to verify correct drive operation.

RADIAL HEAD ALIGNMENT (TRACKING)

The most critical of the tracking adjustments is radial head tracking alignment. Some commercial programs require very accurate tracking in order to read the program and data stored on the disk. This is especially true for programs that have been protected by writing the program in half and quarter track areas of the diskette.

A drive that loads a program from the disk into the system board RAM, but won't read data from tracks on the disk, or that causes the read head to move to the home position several times in the middle of a disk read operation, producing a BDOS error, probably has a tracking problem. The tracking alignment should always be checked after a replacement analog card is mounted on the drive.

To check the alignment of the read/write head, insert the alignment diskette into the drive and set the head to track 16. Connect channel A of your scope to TP1, channel B to TP2, and the external trigger to TP7. Connect the scope's ground to TP6. Set the scope to *add* mode with channel B inverted. Set the sweep time to 20 milliseconds. Set the trigger to positive slope with the voltage input set to 0.2 volt. A "cats-eye" pattern shown in Fig. 4-10 should be displayed on the screen of the scope. Using the technique described below, adjust the radial head alignment screw so the two lobes are as close to equal size as possible but not so one is less than 80 percent of the other.

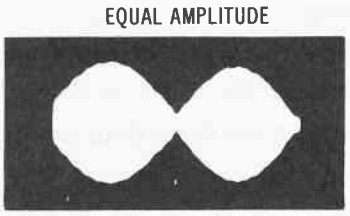


Fig. 4-10. The "cats-eye" screen display.

To Adjust the Radial Head

Tools Required

2 channel scope
Flathead screwdriver
Phillips head screwdriver
Protective pad
Adequate lighting
Tray to hold loose screws
Test diskette
Alignment diskette

1. Turn off power to the computer.
2. Disassemble the computer using the procedures found in Appendix D.
3. Configure the system so the drive to be aligned is set as drive B.
4. Disconnect the disk drive data cable from the back of drive B.
5. Remove the two silver flathead screws holding drive B tight to the chassis, as shown in Fig. 4-4.
6. Gently pull drive B forward about 2 inches out from the front of the chassis.
7. Disconnect the power supply cable from the back of the analog card mounted in drive B.
8. Gently remove drive B out of its chassis.
9. Set drive B on its side so that you can get to the radial head module on the bottom of the drive and connect the scope probes so you can monitor the lobes on the scope.
10. Connect the power supply cable to the back underside of the analog card.
11. Reconnect the disk drive data cable to drive B.
12. Insert the alignment disk in drive B.
13. Boot the test disk in drive A.
14. Set the head on drive B to track 16.
15. Observe the cats-eye pattern for suitability. If the lobes are not within 80 percent of the size of each other, loosen the two head module retaining screws on the bottom of drive B and the one screw on the top next to the radial head alignment screw (see photo on COMPUTER-FACTS page 43).
16. Turn the radial head alignment screw until the lobes are as close to equal as possible.
17. Retighten the three retaining screws.

> Note: Reading how to do this is a lot easier than actually doing it. You need plenty of patience and the steady hands of a surgeon. A slight movement of the module can cause a large change in the size of one of the lobes. Even when retightening the retaining screws, the module has a tendency to shift pulling the alignment out of spec again. Go slowly and be patient.

18. After successfully adjusting the lobes, move the head to track 40, back to track 16, then to track 0 and back to track 16 to make sure the lobe pattern remains within specifications when the head is on track 16.
19. Turn off power to the system.
20. Disconnect all cables to the drive.
21. Carefully push the drive mechanism back into the chassis until about 2 inches of the drive housing is still exposed.
22. Connect the power supply cable to the back underside of the analog card.
23. Push the drive all the way into the chassis housing.
24. Replace the two silver flathead screws on the sides of the drive, as shown in Fig. 4-4.
25. Reconnect the drive data cable to drive B.

26. Power up the computer.
27. Place a *copy* of a program disk in the drive and verify normal write and read operation.
28. Return the drive to service.

INDEX SENSOR ADJUSTMENT

The index sensor is another adjustment that is rarely needed, but if you think this is out of alignment, check its adjustment by connecting the scope channel A to TP3, channel B to TP7, and the ground to TP6. Set the scope's time sweep to 50 microseconds per division and the amplitude to 20 millivolts per division. Set the trigger to channel B (noninverted) on the positive slope. Set the head to track 01 using your test disk and then insert the alignment diskette. Check that the leading edge of the burst occurs 150 to 250 microseconds (best case is 200 microseconds) after the leading edge of the index pulse (Fig. 4-11). If the sensor needs adjustment, take the following actions:

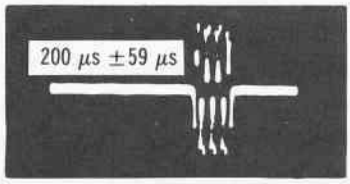


Fig. 4-11. The burst should occur 150 to 250 microseconds after the index pulse.

Tools Required

2 channel scope
Flathead screwdriver
Phillips head screwdriver
Protective pad
Adequate lighting
Tray to hold loose screws
Test diskette
Alignment diskette

1. Turn off power to the computer.
2. Disassemble the computer using the procedures found in Appendix D.
3. Disconnect the disk drive cable from the back of drive B.
4. Remove the two silver flathead screws holding the drive tight to its chassis, as shown in Fig. 4-4.
5. Gently pull the drive forward about 2 inches out from the front of the chassis.
6. Disconnect the power supply cable from the back of the analog card.
7. Gently remove the drive from the chassis.
8. Set the drive on its side so that you can get to the index sensor screw on the bottom of the drive and see the amplitude bursts on the scope.
9. Connect the power supply cable to the back underside of the analog card.
10. Reconnect the drive data cable to drive B.
11. Insert the alignment disk in drive B.
12. Boot the test disk in drive A.
13. Set the head on drive B to track 01.
14. Connect the scope as described in the introductory paragraphs of this section.
15. You should see the same amplitude bursts on the scope that you saw when checking the index sensor setting.
16. Loosen the index sensor retaining screw on the bottom of drive B.
17. Using a flathead screwdriver in the slots on the drive chassis and the sensor (see COMPUTERFACTS page 43), adjust the index sensor until the scope presentation is within specifications (see Fig. 4-11).
18. Retighten the index sensor retaining screw.
19. Turn off power to the system.
20. Disconnect all cables to the drive.
21. Carefully push the drive mechanism back into the chassis until about 2 inches of the drive housing is still exposed.
22. Connect the power supply cable to the back underside of the analog card.

23. Push the drive all the way into the chassis housing.
24. Replace the two silver flathead screws on the sides of the drive, as shown in Fig. 4-4.
25. Reconnect the drive data cable to drive B.
26. Power up the computer.
27. Place a *copy* of a program disk in the drive. Run the system to verify correct operation.

AZIMUTH CHECK

This check verifies the positioning of the head itself. If the azimuth positioning is incorrect, it cannot be adjusted. If this check fails, then the head assembly must be replaced.

To make this check, connect scope channel A to TP1 and channel B to TP2. Connect the external trigger to TP7 and the scope ground to TP6. Set the scope to *add* mode with channel B inverted. The scope sweep time should be set to 0.5 millisecond per division; the voltage amplitude should be set to 0.1 volt per division. Trigger on channel A positive slope. Set the head to track 34 and insert the alignment disk. You should see a pattern like that in Fig. 4-12. If the first and last bursts are not equal to or less than the two middle bursts, the head is bad and should be replaced.

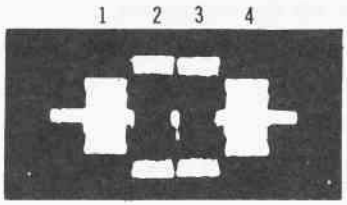


Fig. 4-12. Bursts 1 and 4 should be equal to or less than bursts 2 and 3.

This completes all the preliminary checks that you can conduct on the system before attempting detail troubleshooting checks have been completed, you should proceed to Chapter 5 to trace the failure symptom to a particular area and component. The checks in this chapter are usually enough to find and correct most system failures. Also, by conducting these checks you can eliminate the alignments in this chapter as the problem and use every check that was made as a clue to the failure itself. If the drive still won't read after you successfully completed all the drive checks, you know that the circuitry you checked is not the cause of the problem. Knowing this can lead to other areas not affected by these tests. These areas can then be analyzed to locate the actual cause of the failure. This chapter covered every aspect of preliminary service checks. In Chapter 5, the detailed troubleshooting steps are outlined and described in both classical and flowchart techniques.

5

Detailed Circuit Troubleshooting/Analysis

Chapter 5 covers the detailed troubleshooting analysis associated with the IBM PC. The chapter is divided into five parts as follows:

1. Start-up problems
2. Run problems
3. Display problems
4. Keyboard problems
5. Other I/O problems

Each malfunction in the computer can be associated with one of these areas. A key problem page index covering each of these five general areas is included to provide both classical and flowchart troubleshooting support. Using the index, you can quickly locate a particular problem area for analysis.

Part 1 covers the symptoms that can occur during initial power-up including no power and no disk boot-up operation. Each PC comes with a built-in diagnostic test program and most technicians have a diagnostic disk to use with the built-in diagnostics. It's possible to get a system error number printed on the screen during start-up that helps localize a malfunction to a particular part of the machine. Error codes were described in Chapter 4.

Part 2 discusses the failure symptoms that can occur after initial boot-up, during system operation. These malfunctions include faulty disk read/write operation, bad memory, and program lock-up.

Part 3 addresses difficulties associated with the display portion of the computer. This section includes no display, no text mode, no high resolution or no low resolution, video synchronization failures, character faults, bad graphics, and other problems.

Keyboard problems are detailed in Part 4. This section covers such faults as single and multiple key operation failures, and unwanted repeat action.

Part 5 encompasses other input and output problems, including speaker faults, cassette I/O failures, and light pen malfunctions.

Each part is subdivided into unique failures and provides symptom and repair action or page reference for each circumstance. This data is followed with classical and step-by-step flowchart troubleshooting instructions. Where appropriate, COMPUTERFACTS pages are referenced (for example, CF page 2). Occasionally, in the classical troubleshooting description, a statement is made directing the reader to "check to specs" a

particular part. This means that all the IC input and output pins should be monitored to determine if the correct signals are present on all pins. Sometimes, a flip-flop is being tested. Then, the clock and any preset/preclear inputs should be evaluated with the normal input and output pins.

As you use this manual, you will discover many useful hints for both troubleshooting and repair. Be especially alert to the cautions since further system degradation can occur if you do not follow those procedures exactly as listed.

Before continuing, make sure you've read Chapter 4 and understand the preliminary checks that can be made before dealing with detailed analysis. Few things are as humbling as disassembling a system for troubleshooting and repair, and then discovering that the problem was caused by a bad cable between the IBM system unit and the monitor.

This chapter will not directly discuss malfunctions of components like resistors and capacitors because catastrophic failures of these passive devices are usually quite visible during examination of the system board and peripheral cards. If you feel that a resistor or capacitor has failed, check the component resistance with an ohmmeter. The resistor value should closely approximate that of the code printed on the part (and as listed in the schematic), and the capacitor should not register a short. Most capacitor failures are obvious because caps usually blow apart or char in the circuit. A preliminary check statement referring to "visually examining" a board or card means look for charred, hot, or physically broken parts.

To complement the COMPUTERFACTS component layout photos, the following board layouts will aid in quickly locating possible bad ICs. Figure 5-1 is an IC layout diagram for the IBM PC system board. Figure 5-2 is an IC layout for the color/graphics adapter, and Figure 5-3 shows the board layout of the monochrome display/printer adapter. Figure 5-4 is an IC layout diagram for the disk drive adapter card, and Fig. 5-5 shows the IC layout for the disk drive analog card.

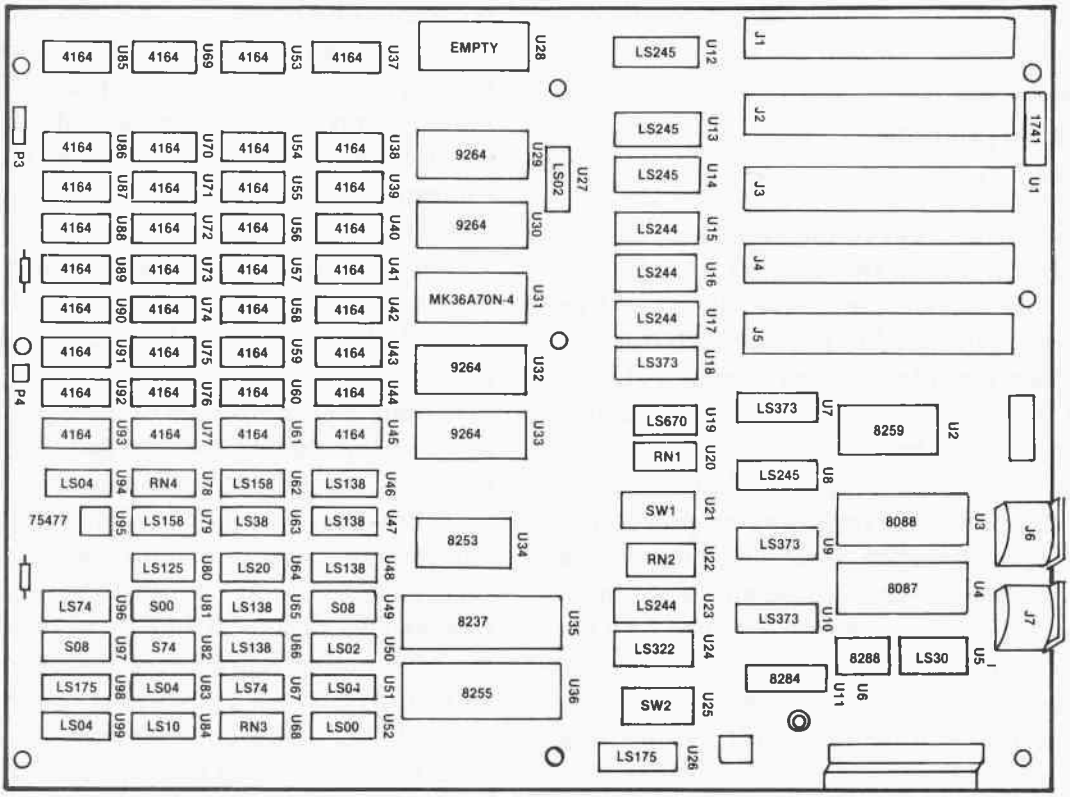


Fig. 5-1. IC layout for IBM PC system board.

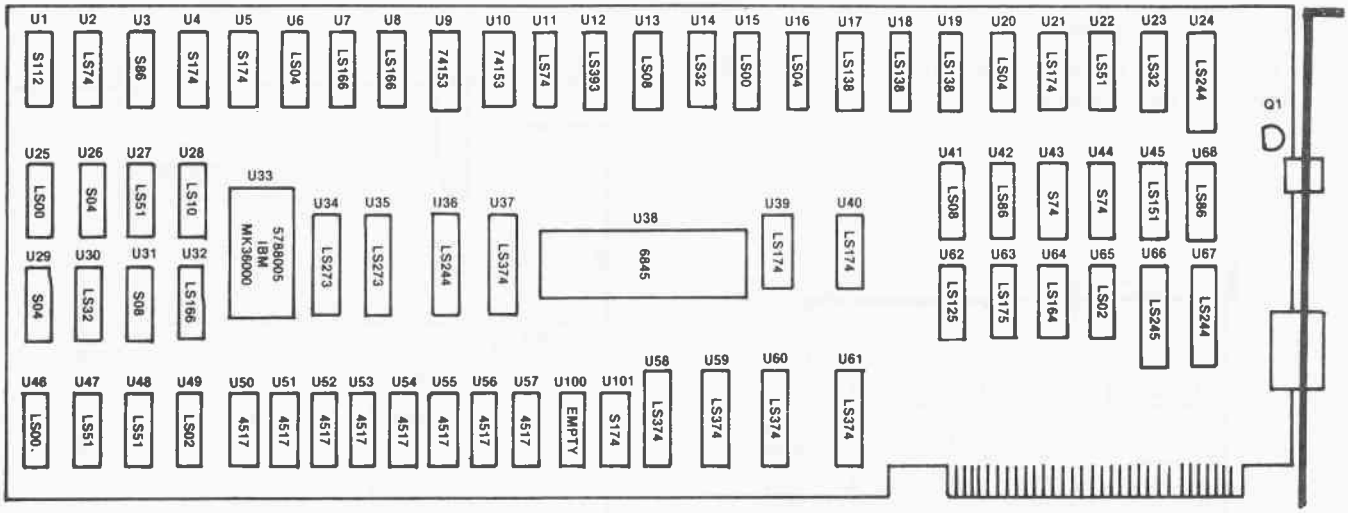


Fig. 5-2. IC layout diagram for color/graphics adapter card.

> Note: The following troubleshooting techniques may require soldering, be sure you are proficient at this process before you proceed.

> **Caution: Opening the IBM PC system unit may void its warranty.**

TROUBLESHOOTING CONTENTS

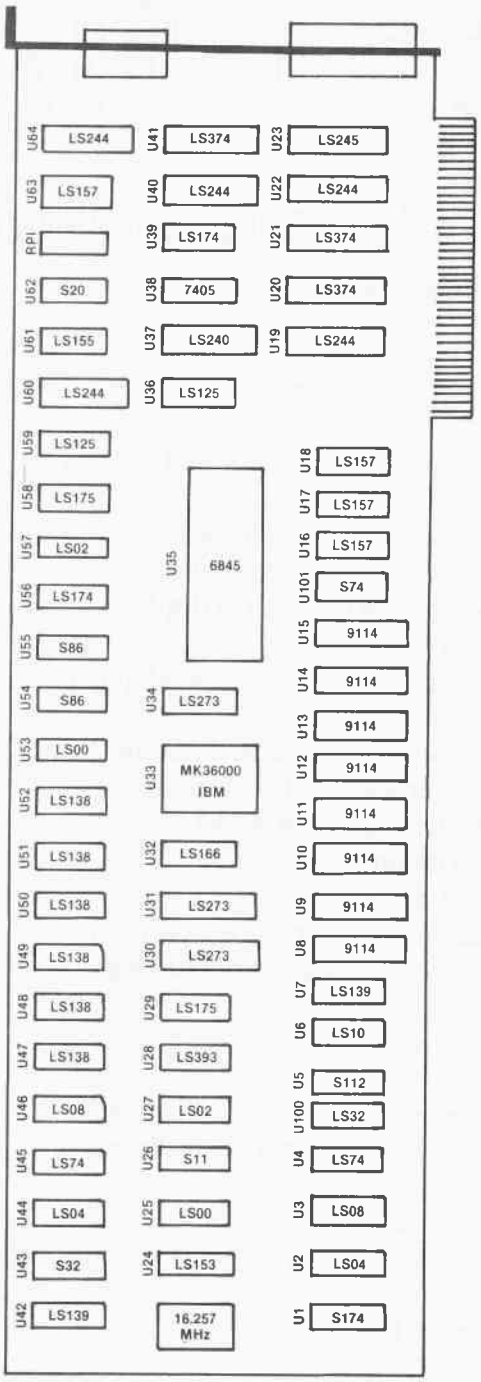


Fig. 5-3. IC layout for the monochrome display/printer adapter.

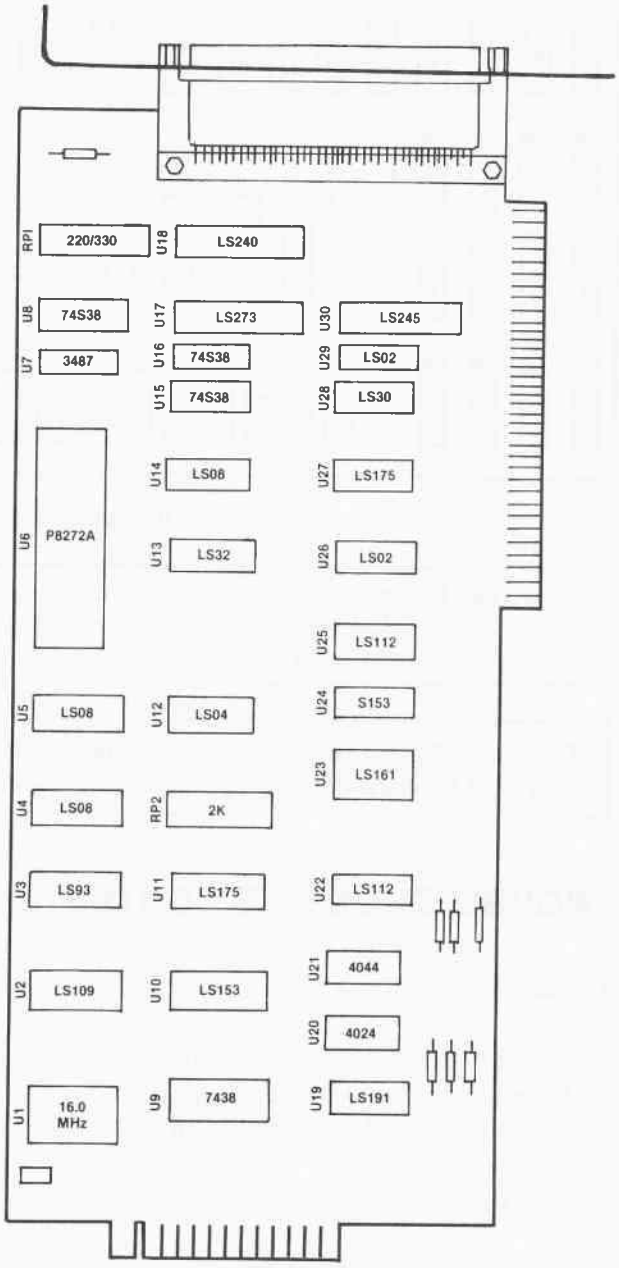


Fig. 5-4. IC layout for disk drive adapter card.

1. IBM PC START-UP PROBLEMS

Four types of error-indications can occur during the initialization or start-up process: beep indicators from the built-in speaker, system error code displays, I/O error code displays, and other error displays. The listing of these codes was covered in Chapter 4. Refer to Chapter 4 if an

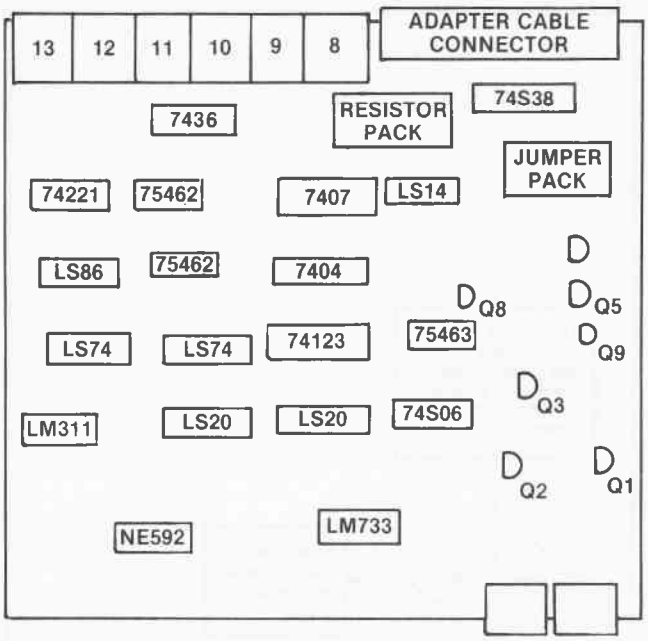


Fig. 5-5. IC layout for disk drive analog card.

error indication occurs. These indications help in isolating a failure to a module, subunit, or peripheral.

> Note: If the system won't boot, the IBM PC DOS manual suggests you reread the manual. You can probably deduce a particular problem faster by noting the conditions of the machine at the time of "failure" and following the troubleshooting steps outlined in this chapter.

Many things can cause the computer to boot improperly or not to boot at all: wrong diskette in the drive, no operating system on the diskette, cables loose, adapter card not fully seated, disk drive failure, memory chip bad, no clock pulses, or even a forgotten unplugged power cord.

Select the subcategory that best describes the symptoms and turn to the appropriate page for troubleshooting.

Problem: Won't Boot, No Fan, Screen Blank

Symptom Described

When the ON/OFF switch is rocked to the ON position, nothing happens. No fan sound can be heard.

Preliminary Checks

1. Check that the external power cable is plugged into the system unit and into a wall socket.
2. Check that the fuse (F1) in the power supply is good.
3. Check the power supply cables to ensure that they are properly seated on the system board (P8, P9).
4. Refer to appropriate section in Chapter 4.

Classical Approach

(Figs. 2-1, 2-2, and CF pages 3, 8, 9, 52, and 53 apply) Remove all the peripherals including the disk drive adapter cards. Power up and note if the fan energizes. If it does, turn off the system and plug each peripheral in one at a time, powering up and testing each time to see if a failure on one of the adapter cards is causing the system to malfunction. When failure occurs, the last card plugged in is bad. Check for shorts between power and ground on the failed board.

If the fan does not turn on after all peripherals are removed, power-down, unplug P8 and P9, and check the voltages in the connectors on the cable from the power supply. If proper power is present on the cable pins, turn off the machine and replug P8 and P9 into the system board. Turn power back on and check pin 1 of P8 for +5 volts. If the logic high is not present, go to (2) in flowchart 5-1. If a logic high is present, check for a short on the system board between power and ground.

If the proper voltages are not present on the cable pins of P8 and P9 when the cables are disconnected from the system board, go to flowchart 5-2. (Also see flowcharts 5-3 and 5-4.)

Problem: Won't Boot, Fan Works, Screen Blank

Symptom Described

No cursor, no screen display, no keyboard response, but can hear fan running in power supply.

Preliminary Checks

1. Check cables for proper mating.
2. Clean all edge connectors on adapter cards.
3. Reseat the CPU chip (U3).
4. Visually inspect the system board.
5. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-5, 2-6, and CF pages 2, 4, and 59 apply)

Remove all peripherals except keyboard, monitor with adapter card. Power-up. Does system boot to ROM BASIC? If it does, turn machine off, and reinstall one adapter card at a time, testing after each installation. When the system fails to boot to ROM BASIC, the last card installed is bad.

If the system doesn't boot to ROM BASIC with all peripheral cards except keyboard and monitor adapter cards installed, check for approximately 14.318 MHz signal on pins 16 and 17 of U11. If the signal is not present, replace crystal X1. If the signal is present, check for a 4.7727 MHz signal on pin 8 of U11. If it is not present, check U11 to specs and replace if bad. If it is present, replace ROM U33 and retest. If the system still fails, check CPU U3, U46, and U6 to specs. Replace if bad. Should this not correct the problem, refer to section "Computer Locks Up, No Keyboard Response" and flowchart 5-5.

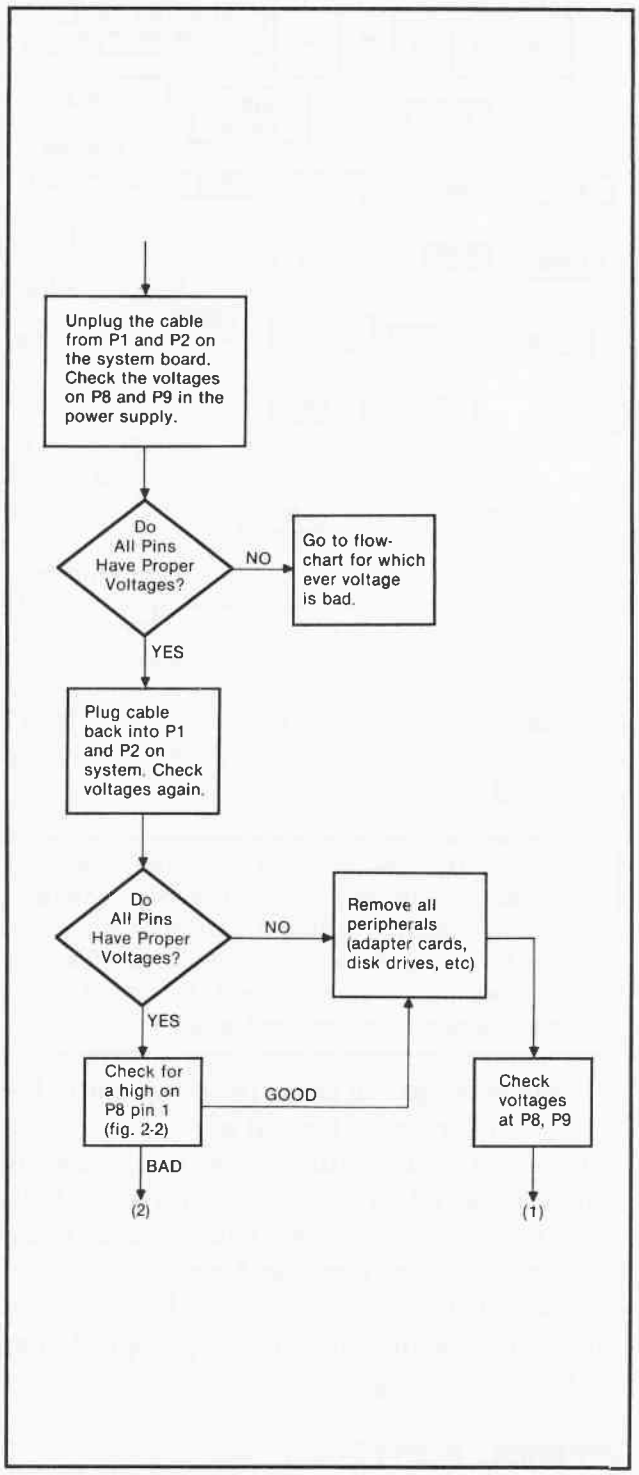
Problem: Won't Boot, Both Drive Lights On

Symptom Described

When system is turned on, drives A and B access indicators light and remain on.

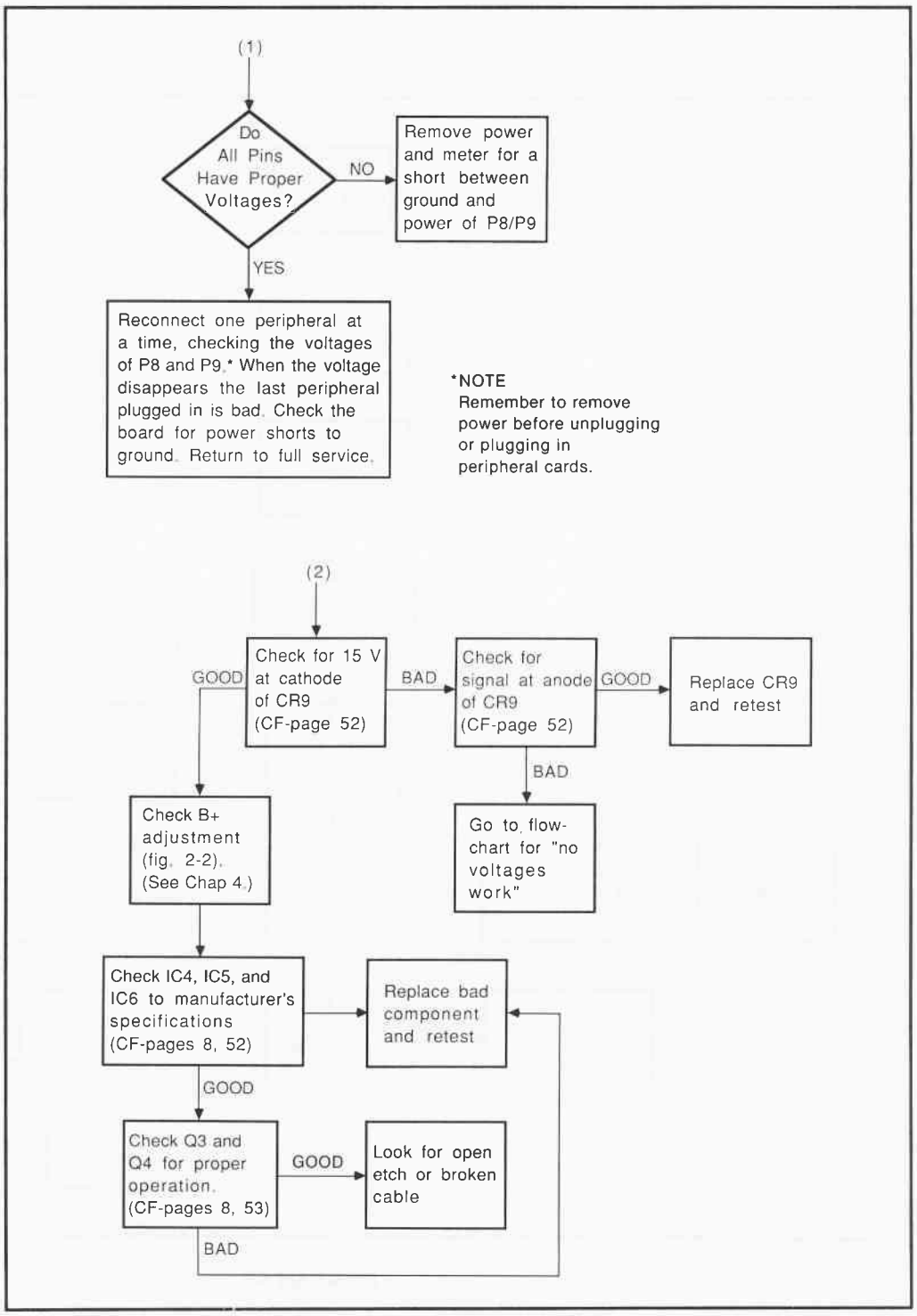
Preliminary Checks (Figs. 2-98 and 2-101 apply)

1. Check that all drive cables are installed correctly.
2. Clean all board header connections, especially on the disk drive adapter card.



Flowchart 5-1.

3. Check that resistor pack at 2F on drive analog card is set for correct drive.
4. Visually inspect the system board, disk drive adapter card, and disk drive analog card.
5. Refer to appropriate section in Chapter 4.



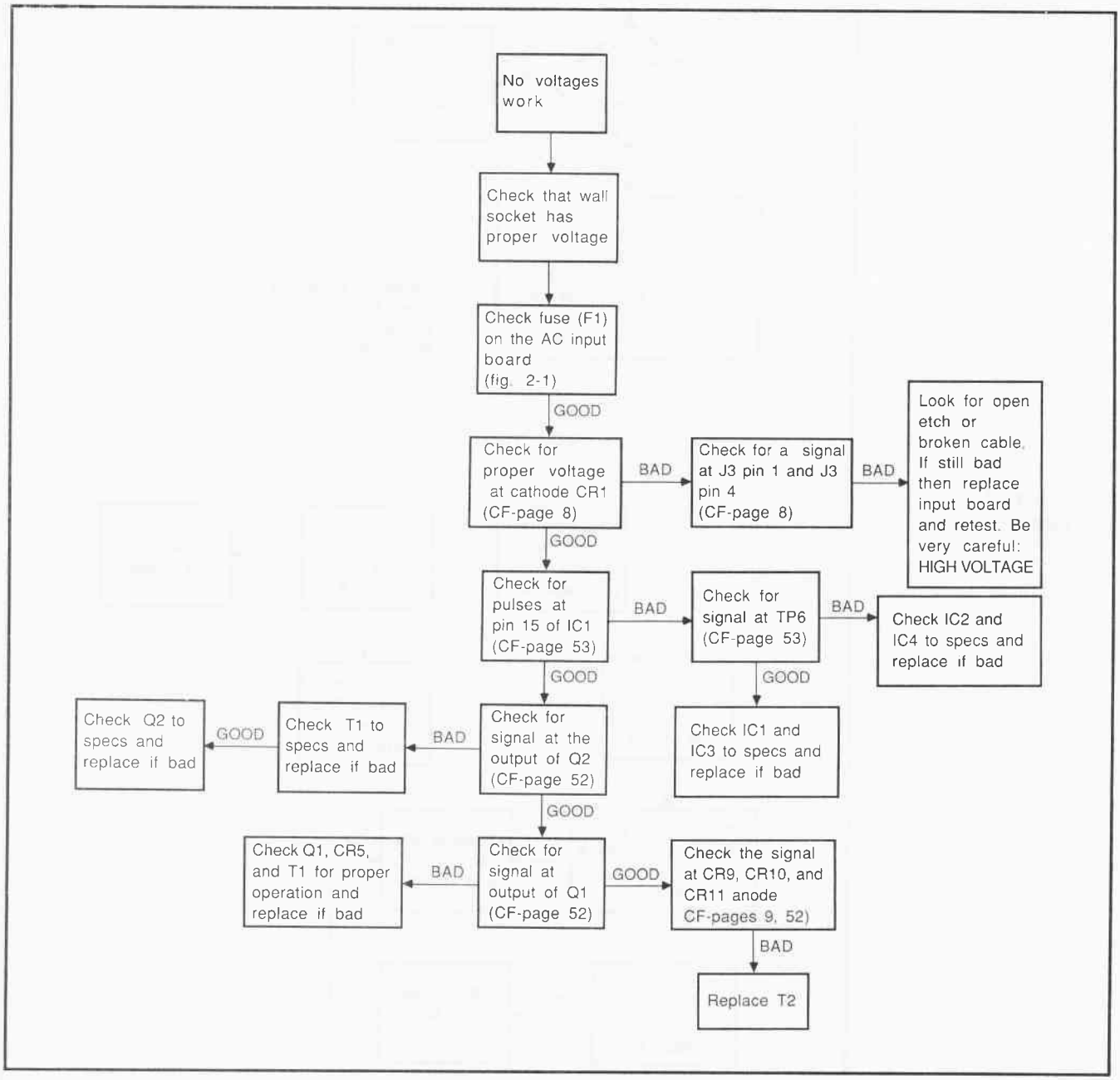
Flowchart 5-1.
"cont."

Classical Approach (Figs. 2-98 and 2-102 apply)

Replace U16 on disk drive adapter card and retest. If this doesn't correct problem, replace U17 and retest. (See flowchart 5-6.)

2. IBM PC RUN PROBLEMS

This section covers those problems you might encounter while your system is running. For example, you attempt to do something and get a



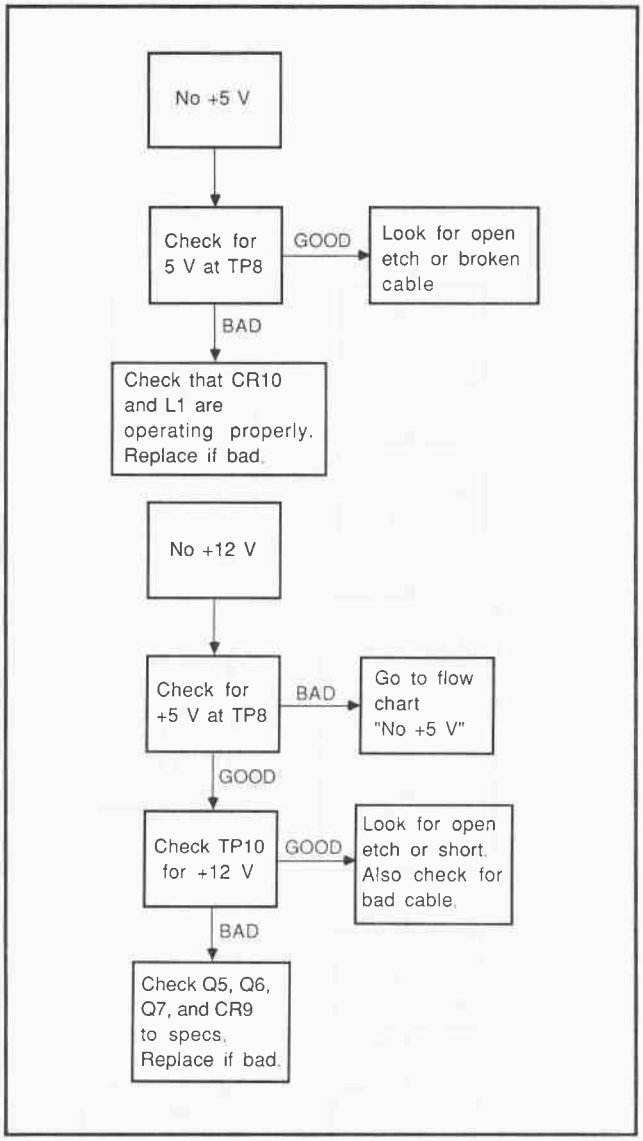
Flowchart 5-2.

response entirely different from what you expected. The following three sections will cover broad malfunctions such as display failure, keyboard failure, and input/output failures (although these may also occur during the time you start up your system).

Problem: Can't Read From One Drive

Symptom Described

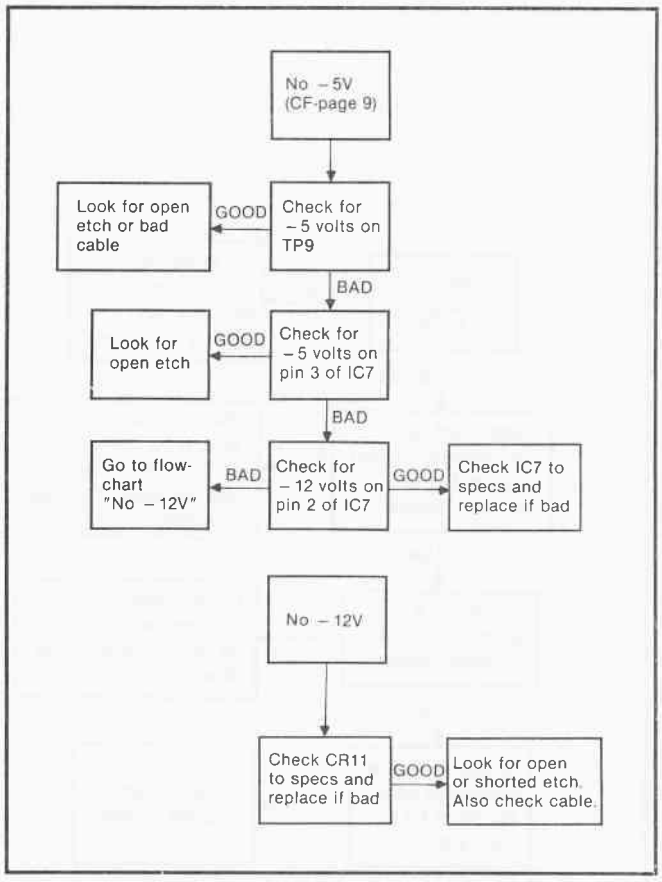
An error display occurs or the drive spins, continuously trying to load data from the disk into the system board.



Flowchart 5-3.

Preliminary Checks

1. Check cables for continuity and proper mating.
2. Try another known good program disk.
3. Shift the bad drive to position A (if not already installed as drive A).
4. Check for disk spinning action during disk read sequence. If no drive motor activation is observed, go to flowchart 5-8.
5. With system turned off, move drive head towards center of disk. Turn system on and note if head moves to track 00. If it doesn't, go to flowchart 5-9.

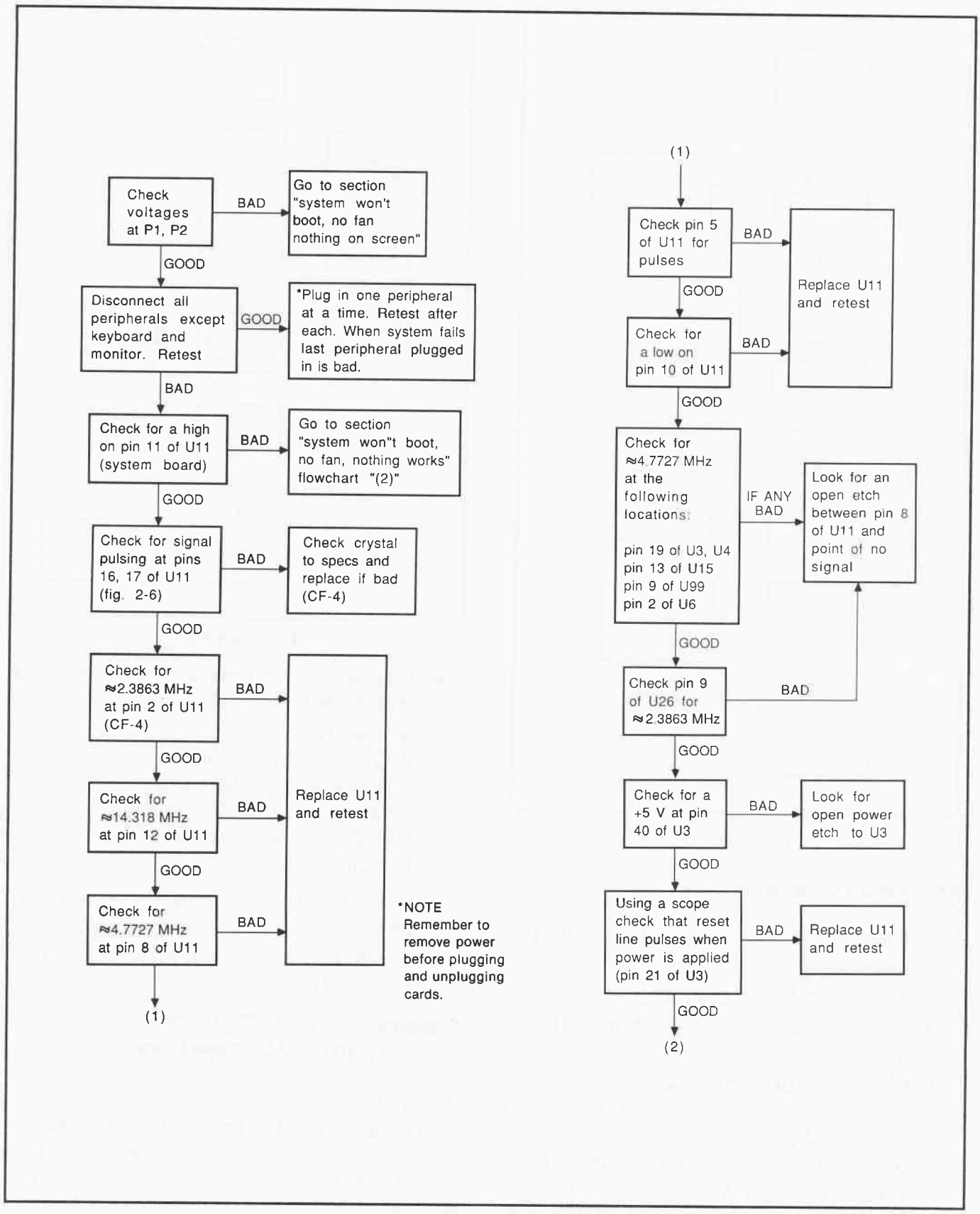


Flowchart 5-4.

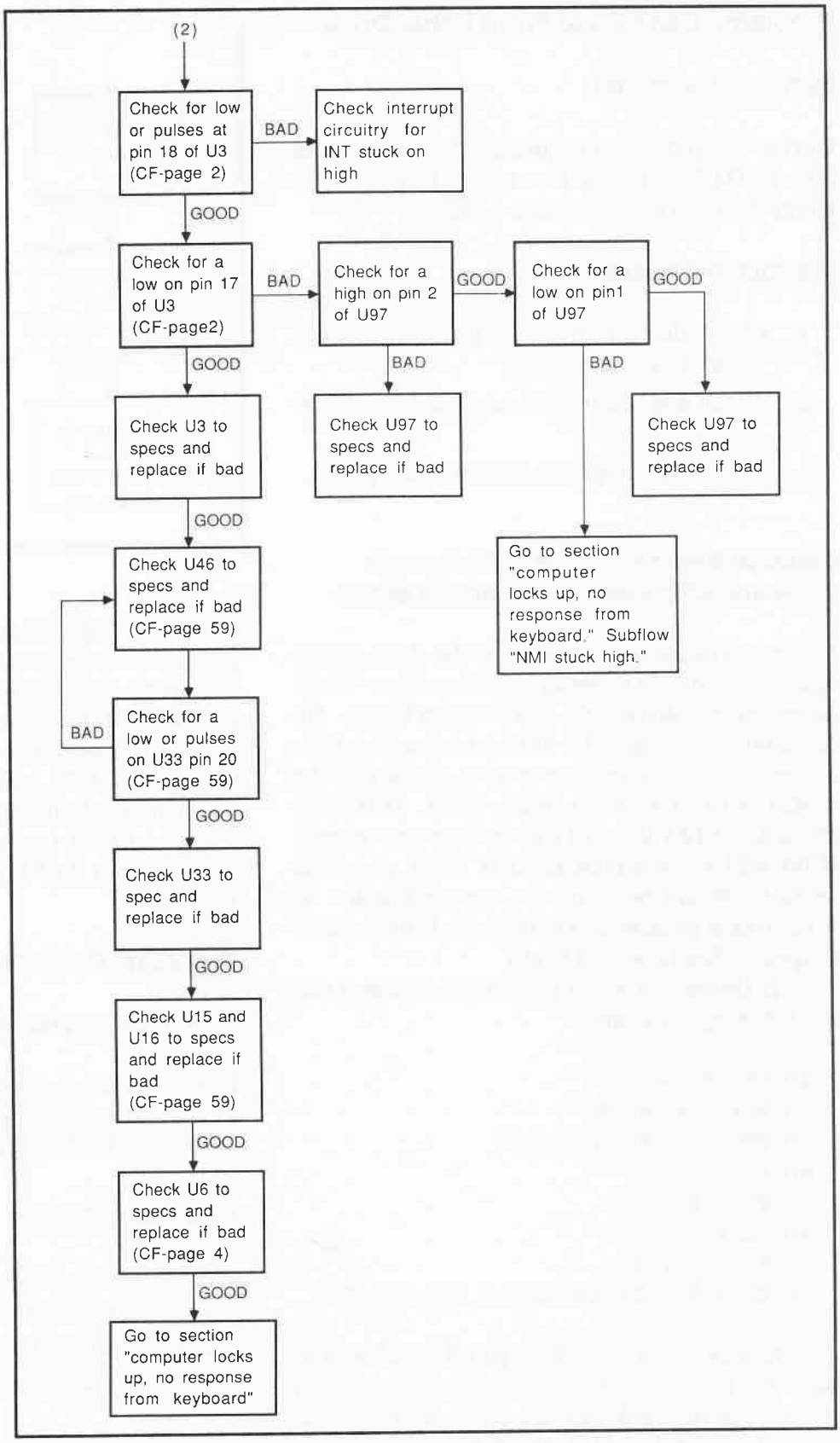
6. Conduct the following adjustments and checks (see Chapter 4).
   A. Speed adjustment
   B. Track 00 adjustment
   C. Track 00 stop adjustment
   D. Radial head alignment
   E. Index sensor adjustment
   F. Azimuth check
7. Check that E1 on analog card is properly jumpered.
8. Clean the drive read/write heads (see Chapter 4).

Classical Approach (Figs. 2-98 through 2-102 and CF page 2 apply)

A drive that will not read is one of the most common failures. Be alert for the clues. Conduct all the preliminary service checks. The flowchart for this problem follows this classical approach to solving the drive won't read problem. (Also see flowcharts 5-7 and 5-10.)



Flowchart 5-5.



Flowchart 5-5. "cont."

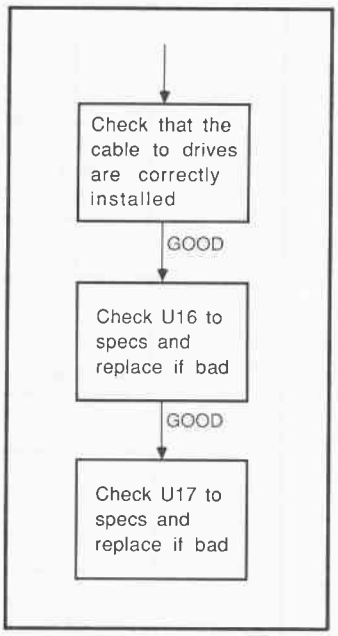
Problem: Can't Read From Either Drive

Symptom Described

During the power-up sequence, the system goes into ROM BASIC. When trying to load from either drive, an error message occurs.

Preliminary Checks

1. Check that all cables are properly connected and have continuity.
2. Clean edge connector on disk drive adapter card.
3. Refer to appropriate section in Chapter 4.



Flowchart 5-6.

Classical Approach (Figs. 2-98 through 2-104 and CF pages 5, 9, 20, and 21 apply)

During a read sequence, note if the drive access light comes on. If it doesn't, go to section "Can't access either drive. No drive access lights. No drive motor energized." If the drive access light comes on during a read operation, check that the motor is rotating the spindle shaft. If it isn't, check for +12 volts on TP10 in the power supply. If no +12 volts is measured at TP10, go to the section "Won't boot, no fan, screen blank." If +12 volts is present at TP10, check U16 and U17 to specs. (See flowchart 5-11.)

If the motor spins properly, write and run the following program:

```
10 DEF SEG=0
20 DEF USR=40000
30 FOR X=40000 TO 40007
40 READ Y
50 POKE X,Y
60 NEXT X
70 A = USR(0)
80 DATA 176,37,186,242,3,238,235,253
```

Check for a signal on pin 8 of U18 (CF page 4). If no signal is present, check for an open or short in the cables to the drive. If a signal is present on pin 8 of U18, check for pulses on pin 23 of U6. If pulses are present, replace U6 and check for the signal on pin 6 of U20 (CF page 20). If no signal is present, check for a signal on pin 7 of U24 (CF page 21).

If the signal is present on pin 6 of U20, check U19, U25, and U26 to specs. If no or a bad signal is found on pin 7 of U24, check U22, U23, and U24 to specs. If a good signal was noted at pin 7 of U24, check U21 and U20 to specs.

Problem: Can't Write to One Drive

Symptom Described

Drive reads properly, but won't write to disk. The error message "DISK WRITE-PROTECTED" may be displayed.

Preliminary Checks

1. Verify no write-protect tab on the diskette being used in the drive.
2. Try a different disk.
3. Check the speed and tracking of the drive in use.
4. Refer to appropriate section in Chapter 4.

Classical Approach (CF page 2 applies)

Configure the bad disk drive as drive B. Insert a nonwrite-protected disk into drive B and close the drive door. Check for a logic low on pin 6 of 3D. If a high is found, check the write protect switch. If pin 6 of 3D is low, place the system in ROM BASIC and write and run the following program:

```
10 OPEN "B:SAM.DAT" FOR OUTPUT AS #1
20 FOR X = 1 TO 300
30 PRINT #1, "THIS IS A TEST"
40 NEXT X
50 CLOSE #1
60 GOTO 10
```

Check for pulses on pin 8 of 3B. If no pulses are found, check 3B to specs. Follow any improper signal back to its origin. If 3B has proper pulses, check the waveform on the collector of Q1 and Q2 (CF page 2). If a good waveform is noted, check the resistance of the heads and the operation of Q6 and Q7.

If the waveform is bad, check for pulses on pins 8 and 9 of 5C. If the pulses are present, check 2B to specs and replace if bad. If 2B is good, check Q1 and Q2 to specs. If bad pulses are noted on pin 8 or pin 9 of 5C, check 2E to specs and replace if bad. Check Q3 to specs and replace if bad. Go to the section "Can't read from one drive," subflowchart "Head Switching Bad" in flowchart 5-12.

Problem: Can't Write to Either Drive

Symptom Described

When writing to either drive, an error message occurs and the operation ends, or the data that was written to the disk is not present during a later read operation.

Preliminary Checks

1. Check all cables for continuity and proper mating.
2. Clean disk drive adapter card edge connector.
3. Configure system as single drive, and test each drive individually.
4. Verify no write-protect tabs on disks being used.
5. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-98 through 2-103 and CF pages 4 and 20 apply)

Insert a blank, nonwrite-protected disk in drive B. Go into ROM BASIC and write and run the following program:

```
10 OPEN "B;SAMS.DAT" FOR OUTPUT AS #1
20 FOR X = 1 TO 300
30 PRINT #1, "THIS IS A TEST"
40 NEXT X
50 CLOSE #1
60 GOTO 10
```

Replace U6 and retry writing. If this doesn't work, check for a low on pin 16 of U18. If a high is measured, check for a high on pins 4 and 15 of U18. If pins 4 and 18 are high, replace U18. If pins 4 and 18 are low, check the cable for a short or an open (CF page 4). If pin 16 of U18 is low, check for pulses on pin 3 of U9. If no pulses are noted, check U9, U10, and U11 to specs. If there are pulses on pin 3 of U9, check P2 continuity. (See flowchart 5-13.)

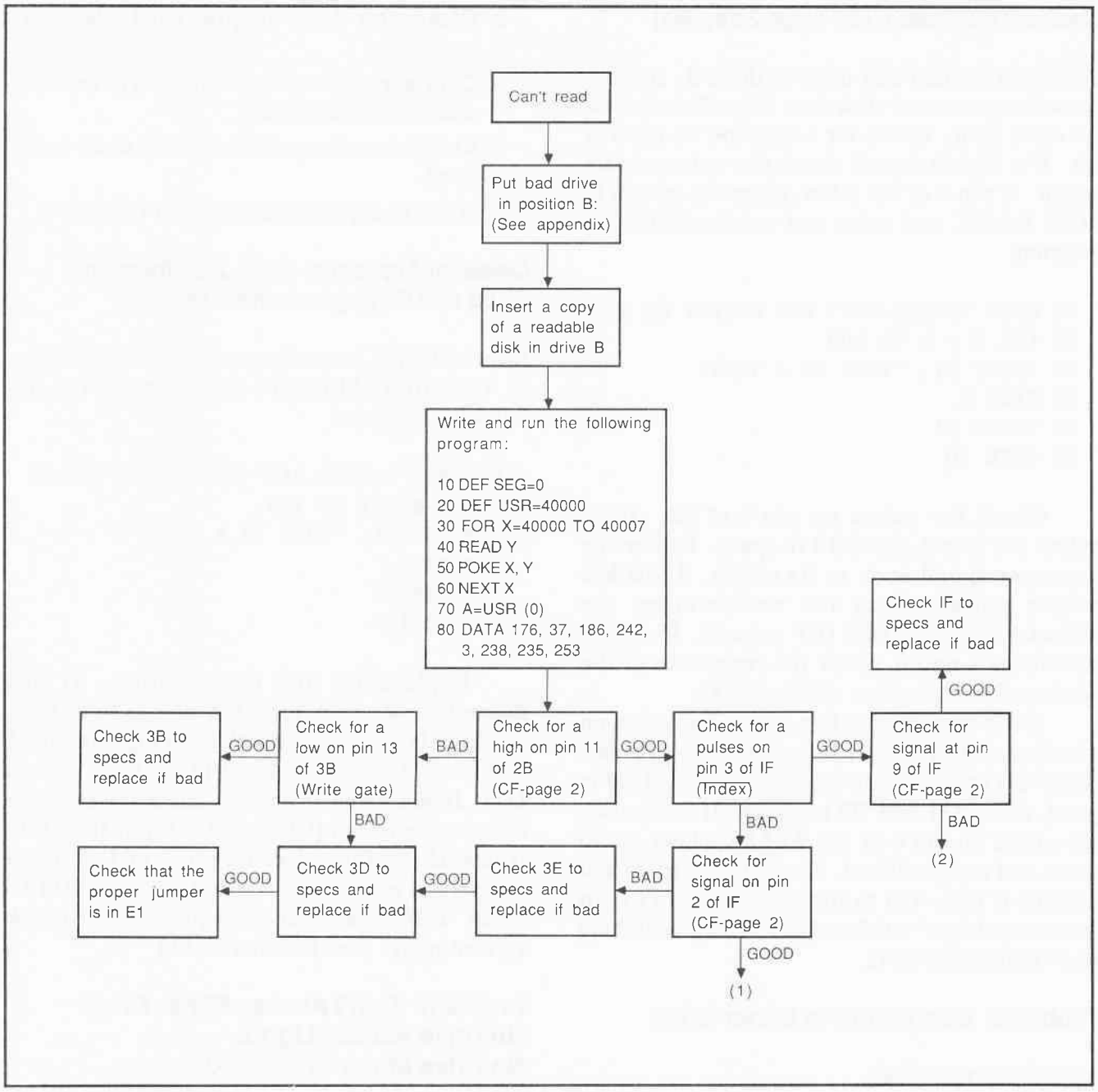
Problem: Can't Access Either Drive. No Drive Access Lights. No Drive Motor Energized

Symptom Described

Both drives fail to react to any read or write operation.

Preliminary Checks

1. Check that all drive data and power cables are properly connected and have good continuity.



Flowchart 5-7.

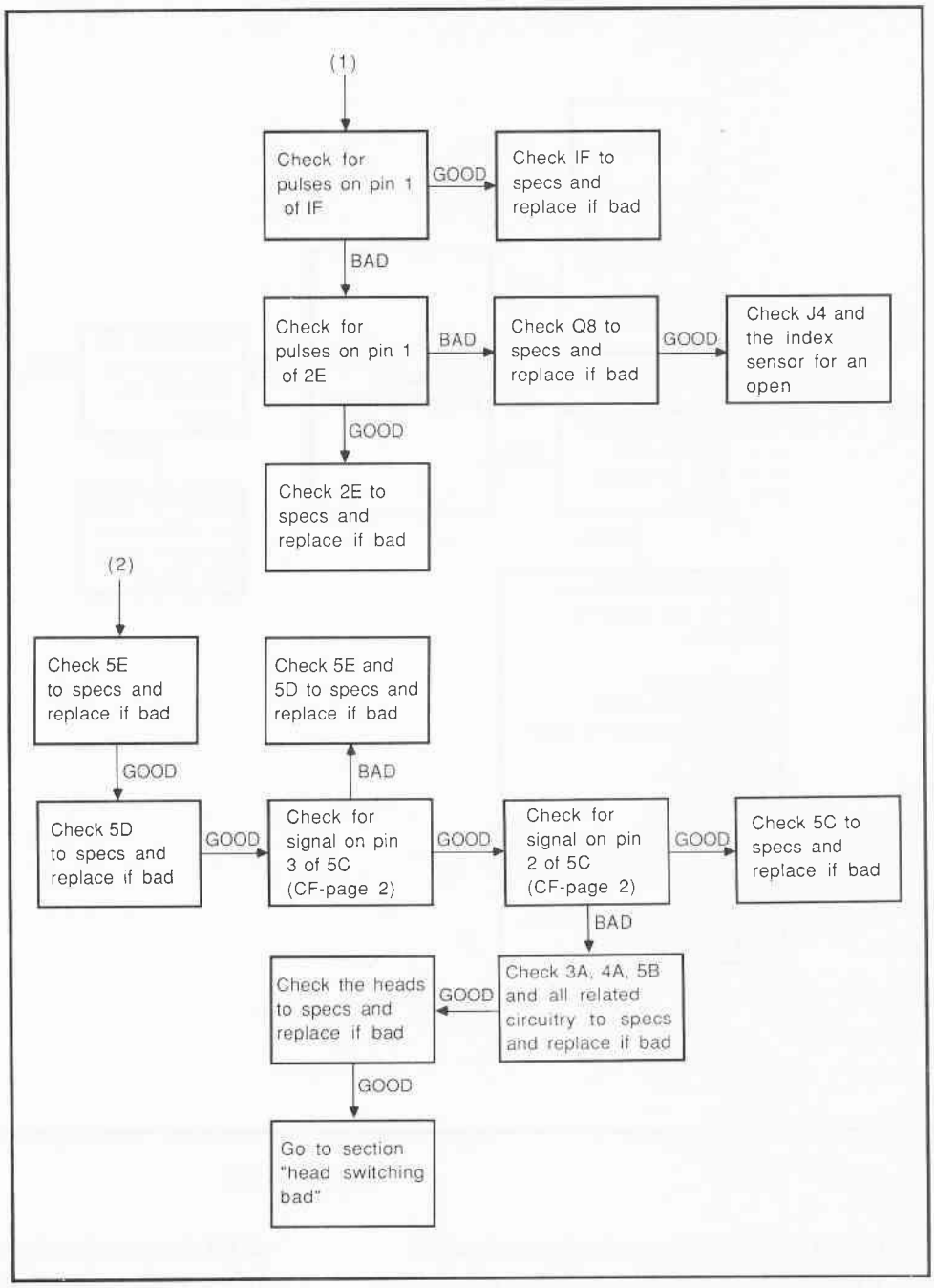
2. Clean edge connector on drive adapter cards.
3. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-98 and 2-101 apply)

Go into ROM BASIC and write and run the following program:

```
10 DEF SEG=0
20 DEF USR=40000
30 FOR X = 40000 TO 40007
40 READ Y
50 POKE X,Y
60 NEXT X
70 A=USR(0)
80 DATA 176,37,186,242,3,
   238,235,253
```

Flowchart 5-7. "cont."



Check for a low on pin 6 of U16. If pin 6 is low, check the cables on P2 for continuity. If pin 6 is high, check U30 and U17 by replacing each. If neither of these is bad, check U29, U14, and U16 to specs and replace if bad. (See flowchart 5-14.)

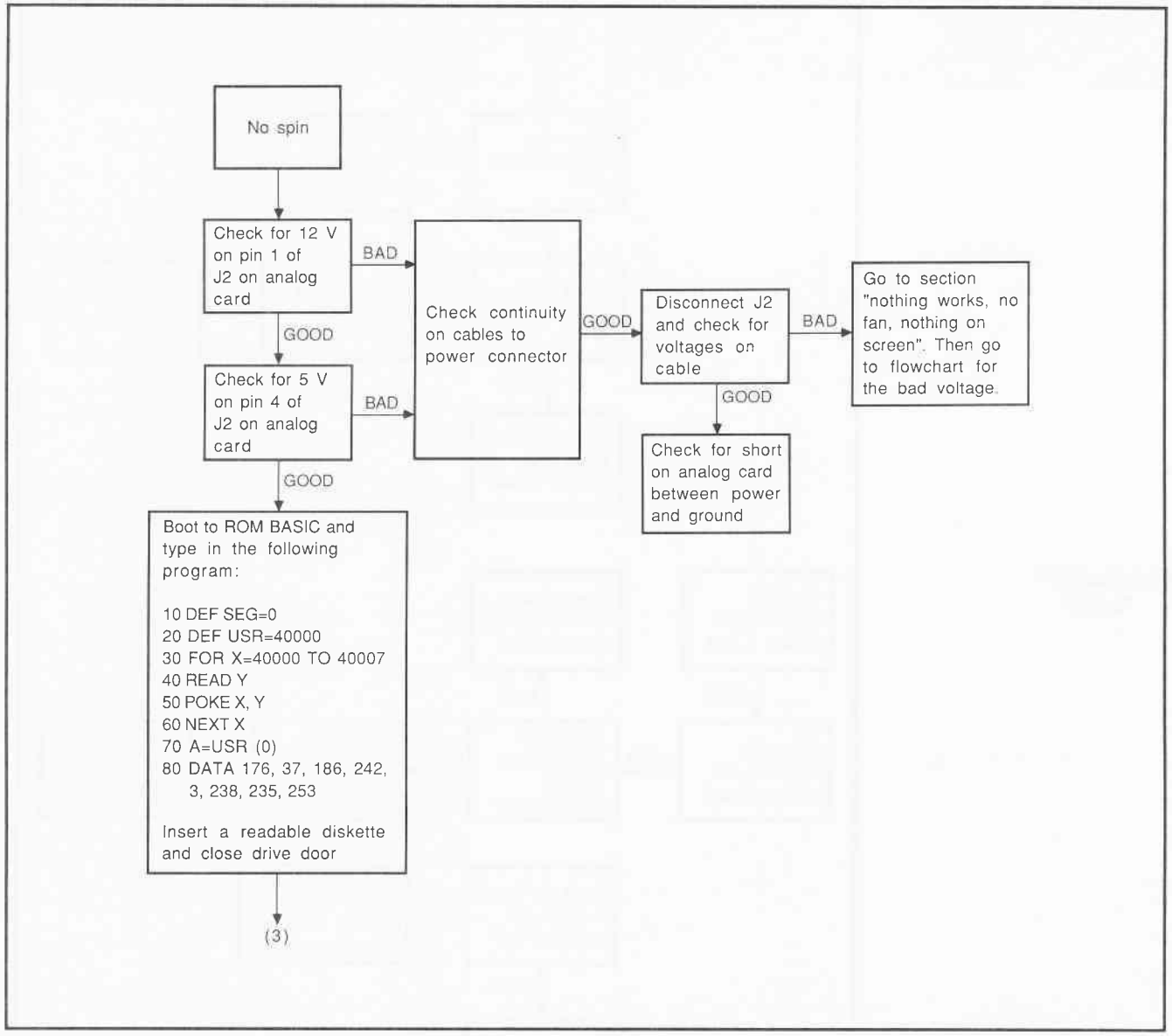
Problem: Drive Destroys Data on Write-Protected Disk

Symptom Described

When inadvertently trying to write to a disk which has a write-protect tab installed, data is written on the disk. The write-protect feature is not functioning.

Preliminary Checks

1. Check the drive cables for proper mating and continuity.



Flowchart 5-8.

2. Clean the adapter card edge connector.
3. Refer to appropriate section in Chapter 4.

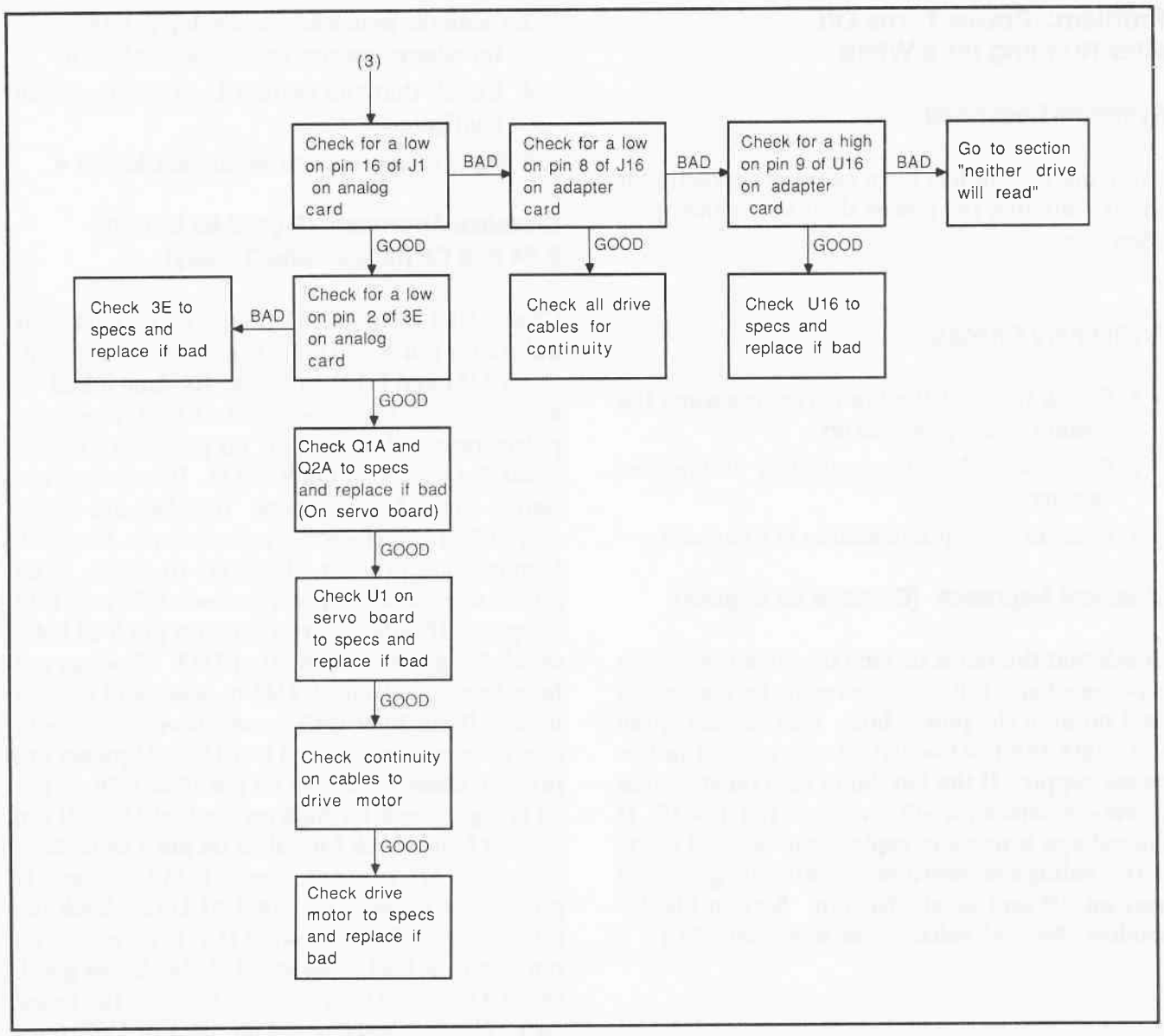
Classical Approach (CF page 2 applies)

Insert a write-protected blank disk into the bad drive and close the drive door. Check for a low on pin 10 of 3D (analog card). If pin 10 is high, check for a high on pin 11 of 3D. If pin 10 is low, check 3B to specs and replace if bad. If pin 11 of 3D is high, replace 3D. If pin 11 of 3D is low, check the write protect switch (check pins 1 and 2 of P8 for continuity when the disk is not in the drive, and open when a write-protected disk is inserted). (See flowchart 5-15.)

Problem: While Running, Computer Locks Up. No Keyboard Response

Symptom Described

During operation of known-good program, the computer locks up, the display freezes, and keyboard inputs have no effect.



Flowchart 5-8. "cont."

Preliminary Checks

1. Try a different program disk.
2. Check all cables for continuity and proper mating.
3. Clean edge connectors on adapter cards.
4. Reseat the CPU (U3) on the system board.
5. Refer to appropriate section in Chapter 4.

Classical Approach: (Figs. 2-7, 2-13, 2-15, and CF pages 2, 3, 4, 58, and 60 apply)

This problem can be caused by a failure in the non-maskable interrupt section, the ready circuitry, the bus control circuitry, the reset circuitry, the timing circuitry, or the processor circuitry. The flowchart follows the classical approach. (See flowcharts 5-16 through 5-20.)

Problem: Power Turns Off after Running for a While

Symptom Described

After the system has been running correctly for about 1 minute, the power shuts down and operation stops.

Preliminary Checks

1. Check to see if the fan is running when the system is first powered up.
2. Check all cables for continuity and proper mating.
3. Refer to appropriate section in Chapter 4.

Classical Approach (CF page 53 applies)

Check that the fan is turning on when the system is powered up. If it spins correctly, let the system heat up until the power fails. Use coolant spray to isolate the heat sensitive component in the power supply. If the fan didn't turn on at system power-up, check for -12.2 volts on pin 1 of J5. If the voltage is present, replace the fan and retry. If the voltage is improper or missing, go to the section "Won't boot. No fan. Screen blank" subflow "No +15 volts." (See flowchart 5-21.)

3. IBM PC DISPLAY PROBLEMS

Problem: No Display (Monochrome Adapter)

Symptom Described

No graphics or text characters can be produced on the display monitor.

Preliminary Checks

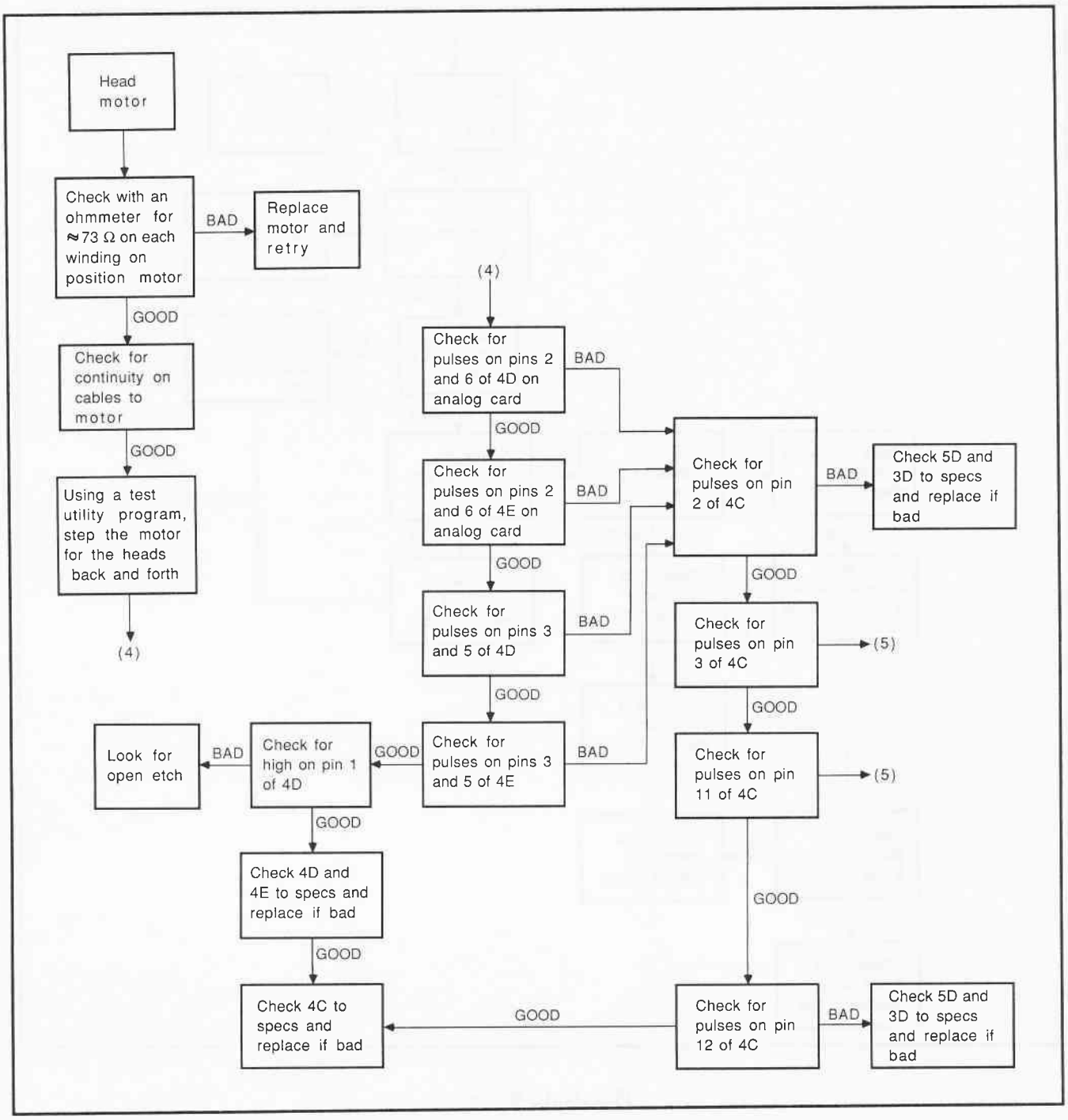
1. Check the video cable for continuity and proper mating.
2. Try another monitor.
3. Clean the monochrome display/printer adapter edge connector and reseat the board.
4. Check that the system is properly switch configured.
5. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-65 through 2-84 and CF pages 2 and 3 apply)

Check U64 to specs. If U64 is good, check for pulses on pin 8 of U43. If pulses are present, check U54 and U101 to specs. Replace if bad. If no pulses are found on pin 8 of U43, check for pulses on pin 9 of U43. If no pulses are found, check for pulses on pin 9 of U3. If no pulses are found, check U28 to specs. If pulses are found on pin 9 of U3, check for pulses on pin 10 of U3. If pulses are present, check U3 to specs. If no pulses are found on pin 10, check U55 and U35 to specs. If pulses were found on pin 9 of U43, check for pulses on pin 10 of U43. If pulses are found on pin 10, check U43 to specs and replace if bad. If pin 10 of U43 doesn't have pulses on it, check for pulses on pin 11 of U26. If pulses are present, check for a high on pin 10 of U26. If pin 10 is high, check for high on pin 9 of U26. If pin 9 is not high, check for pulses on pin 1 of U62. If no pulses are present, check U28 to specs. If pulses are present on pin 1 of U62, check for pulses on pin 5 of the same IC. If no pulses on pin 5, check U44 to specs. If U44 checks good, check U35 and U55 to specs. Replace the failed part. If pulses were noted on pin 5 of U62, refer to (3) on flowchart 5-22.

If pin 10 of U26 is not high, check for pulses on pin 8 of U27. If pin 8 is pulsing, check for pulses on pin 9 of U27. If pulses are noted on pin 9, check U27 and U29 to specs. Replace the failed component. If pin 8 of U27 is not pulsing, check U48 to specs and replace if bad. If U48 checks good, refer to the flowchart, subsection 7.

If pin 9 of U27 is not pulsing, check U49 to specs. If U49 checks good, refer to (7) on flowchart 5-22. If pin 11 of U26 is not pulsing, check for a low on pin 13 of U43. If pin 13 is low, check for pulses on pin 12 of U43. If pulses are present, replace U43. If pulses are not on pin 12, check U32 to specs. If U32 checks good, check



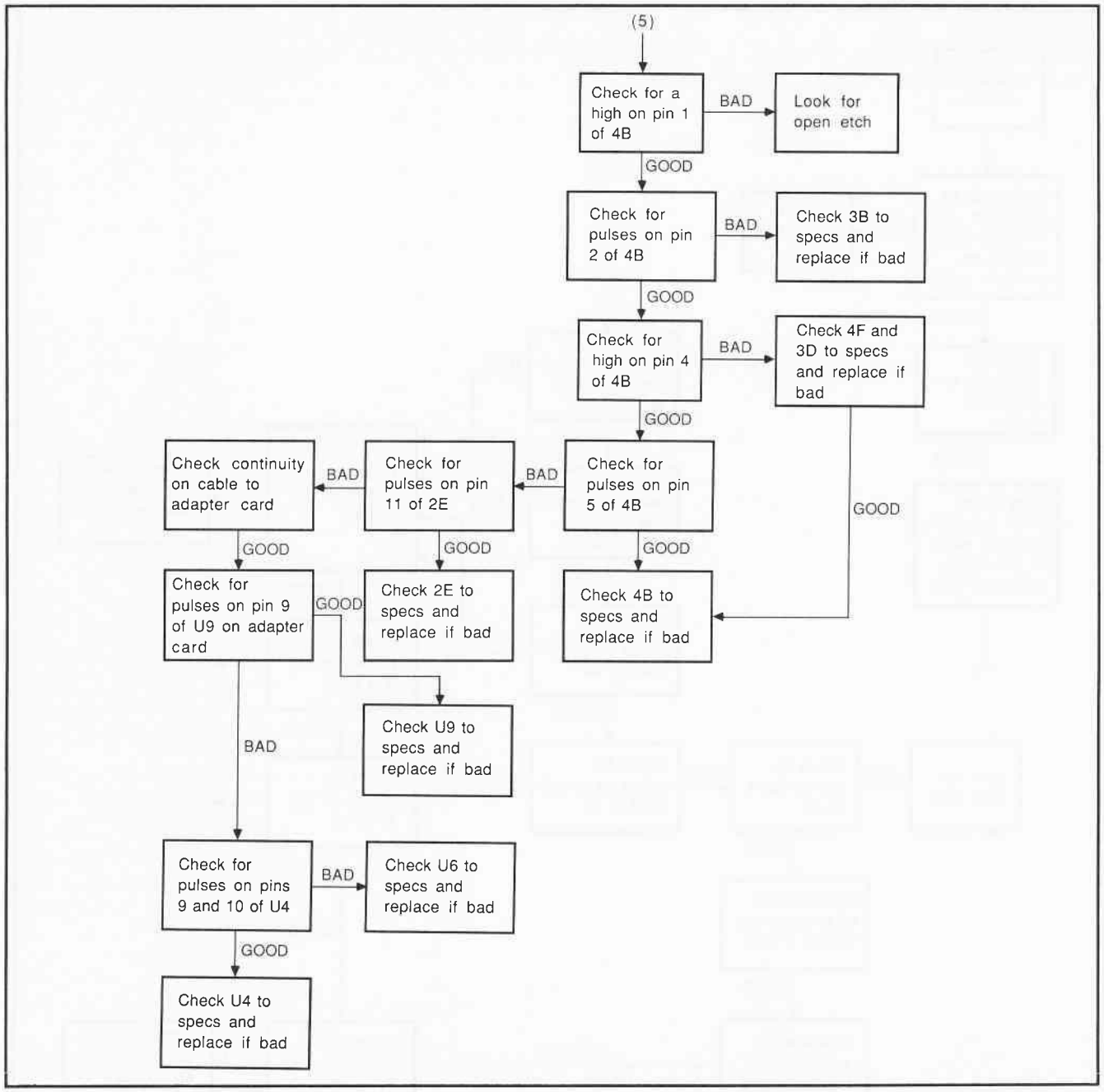
Flowchart 5-9.

U33 to specs. If U33 also checks good, refer to (4) of flowchart 5-22. If pin 13 of U43 is not low, check for a low on pin 12 of U29. If pin 12 is high, refer to (5) of flowchart 5-22. If pin 12 is low refer to (2) of flowchart 5-22.

Problem: No Horizontal Sync (Monochrome Adapter)

Symptom Described

The display is unreadable due to a lack of horizontal sync signals.



Flowchart 5-9. "cont."

Preliminary Checks

1. Check the video cable for continuity and proper mating.
2. Clean the monochrome adapter card edge connector pins.
3. Try a different, known-good monitor.
4. Refer to appropriate section in Chapter 4.

Classical Approach (Fig. 2-80 and CF pages 2, 3, and 15 apply)

Check for a signal on pin 13 of U64 (CF page 3). If a signal is found, check U64 to specs and replace if bad. If no signal is on pin 13, check for

a signal on pin 13 of U55 (CF page 2). If no signal is found, check U35 by substitution. If a signal is found on pin 13, check for the signal on pin 5 of U3 (CF page 3). If no signal is on pin 5, check U55 to specs. If U55 checks good, check U100 to specs. If a good signal is noted on pin 5 of U3, check for a high on pin 4 of U3. If pin 4 is high, replace U3. If pin 4 is low, check for a high on pin 10 of U58. If pin 10 is high, check for a high on pin 1 of U45. If pin 1 is high, check U45 to specs. If pin 1 isn't high, check U56 to specs. If pin 10 of U58 is low, check U50 and U60 to specs. Replace the failed part. (See flowchart 5-23.)

Problem: No Vertical Sync (Monochrome Adapter)

Symptom Described

The display continues to roll due to lack of a vertical sync signal.

Preliminary Checks

1. Check the video cable for continuity and proper mating.
2. Clean the monochrome adapter card edge connector pins.
3. Try a different known-good display monitor.
4. Refer to appropriate section in Chapter 4.

Classical Approach (Fig. 2-80 and CF pages 2 and 3 apply)

Check for signal on pin 14 of U55 (CF page 2). If the signal is not present, check U35 by substitution. If a signal is present on pin 14, check for signal on pin 11 of U64 (CF page 3). If a signal is found, check U64 to specs. If pin 11 of U64 has no signal present, check for a high on pin 2 of U54. If pin 2 is low, check that there is no jumper at J1. If pin 2 is high, check U54 to specs. (See flowchart 5-24.)

Problem: No Low or High Resolution Display (Monochrome Adapter)

Symptom Described

The text works, but there is no low or high resolution graphics display capability.

Preliminary Checks

1. Check all cables for continuity and proper mating.
2. Clean the adapter card edge connector pins.
3. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-72, 2-76, 2-80, and CF page 2 apply)

Run a program that activates the high resolution screen. Check U58 to specs. If U58 is good, check for a low on pin 2 of monochrome adapter card IC U24. If pin 2 is low, check U24 to specs. If pin 2 is not low, check for a high on pin 6 of U53. If pin 6 is high, check U6 to specs. If pin 6 is not high, check U53 to specs. (See flowchart 5-25.)

Problem: Bad Characters (Monochrome)

Symptom Described

Garbage on screen, illegal characters, strange shapes in both graphics and text modes.

Preliminary Checks

1. Verify cables connected correctly.
2. Clean adapter card edge connector pins.
3. Check that system is configured properly.
4. Try different known-good monitor.
5. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-76, 2-79, 2-80, and CF pages 2 and 3 apply)

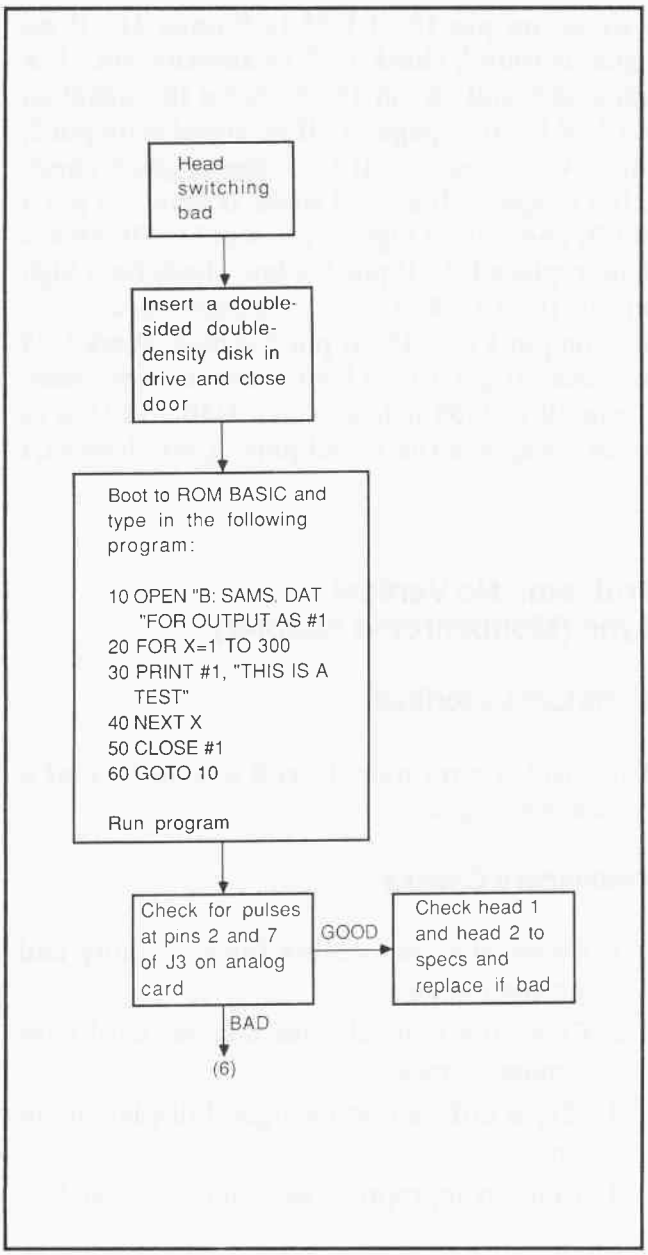
Check U33 by substitution. If problem persists, check U64 to specs. If U64 is good, check for signal on pin 4 of U54 (CF page 2). If a good pin 4 signal is noted, check for pulses on pin 5 of U54. If pulses are present, check U54 and U101 to specs. If no signal was found on pin 4 of U54, check U32 to specs. If U32 is good, check for signal on pin 10 of U43. If the pin 10 signal is good, check for pulses on pin 9 of U43. If pulses are found, check U43 to specs. If no pulses are found, check U35 to specs. If U35 checks good, check U3 to specs. If U3 checks good, check U55 to specs. If U55 checks good, check U28 to specs.

If a bad or no signal was found on pin 10 of U43, check for signal on pin 11 of U26. If a good signal is found, check U26 to specs. If a bad signal is noted, check U43 to specs. If U43 is good, refer to (4) of flowchart 5-26.

If pin 5 of U54 is bad, check for pulses on pin 5 of U29. If pulses are found, check for pulses on pin 9 of U29. If pulses are found, check for pulses on pin 1 of U29. If pulses are found, check U29 to specs. If no pulses are found on pin 5 of U29, check U48 and U49 to specs. If these ICs check good, check U27 to specs.

If no pulses were found on pin 9 of U29, check U100 to specs. If U100 checks good, check U1 and U6 to specs.

If no pulses were found on pin 1 of U29, check U35 by substitution. If U35 checks good, check U43 to specs. If U43 checks good, check U55 to specs. If U55 is good, check U23 to specs.



Flowchart 5-10.

Problem: No Display (Color Graphics Adapter)

Symptom Described

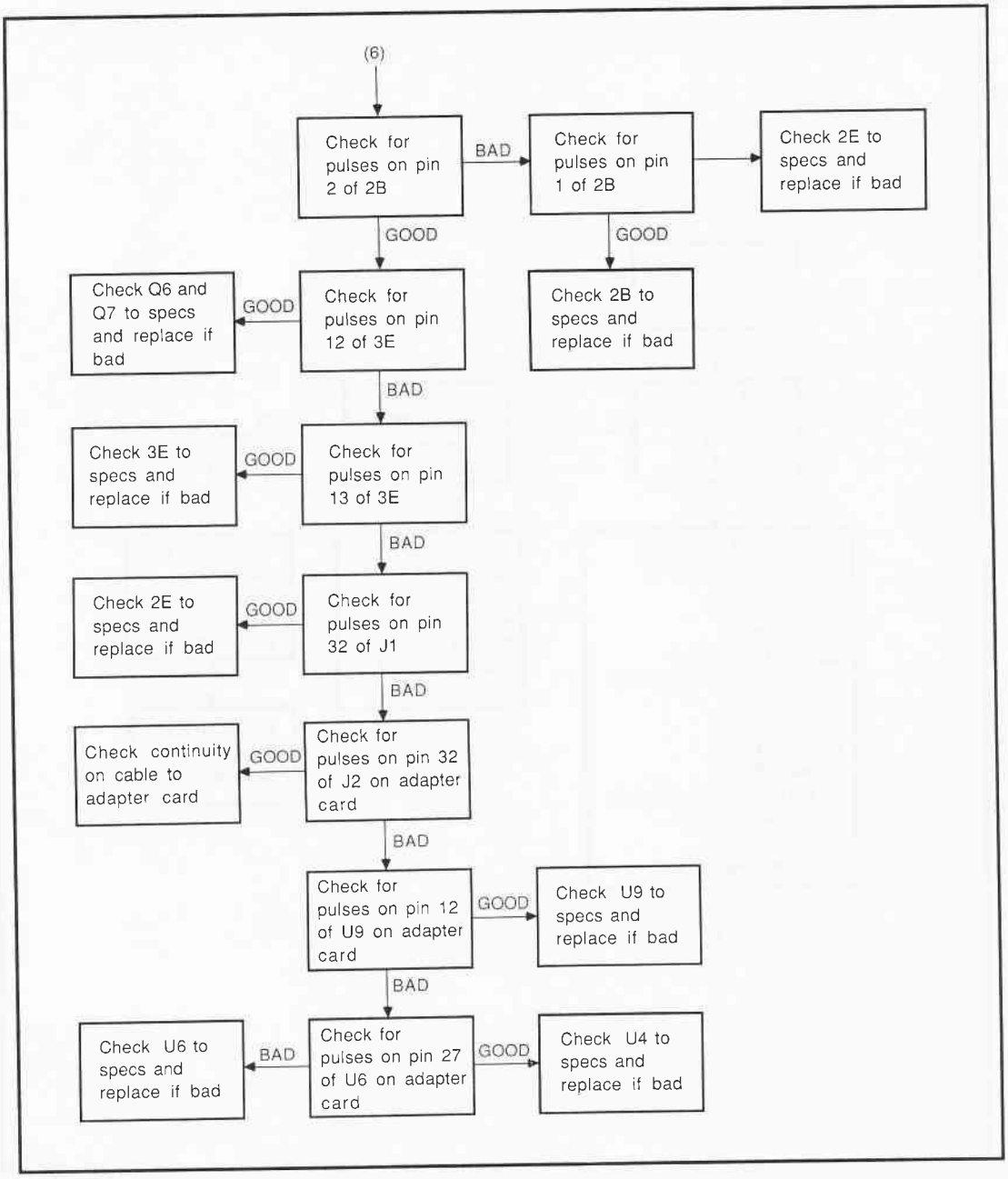
No screen display. No graphics; no text.

Preliminary Checks

1. Check the cable for continuity and proper mating.
2. Try another known-good display monitor.
3. Clean the adapter card edge connector pins.
4. Check for proper system configuration.
5. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-91, 2-94, 2-95, and CF pages 16 and 17 apply)

Does the direct drive video work? If it does, check Q1 to specs. Look for an open between



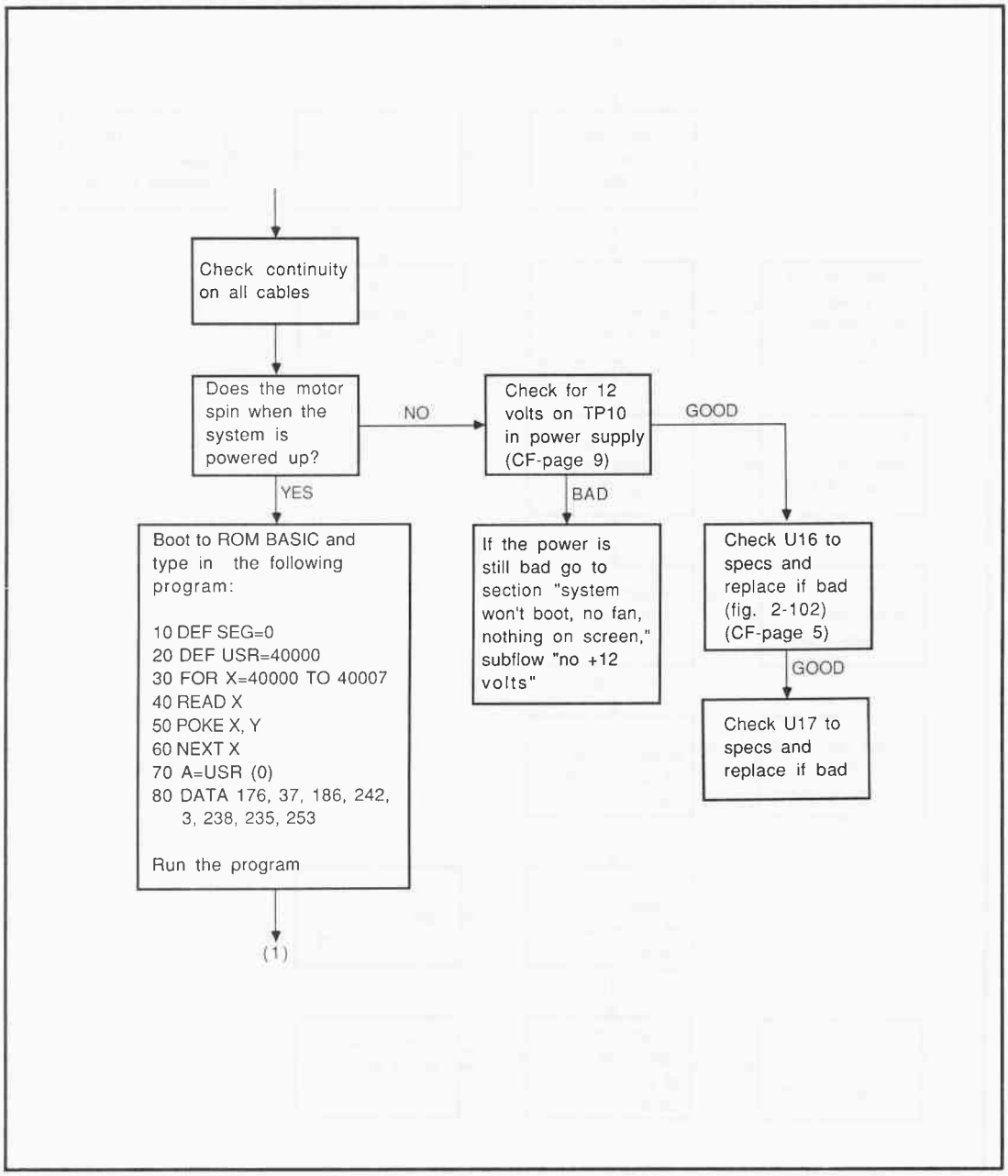
Flowchart 5-10. "cont."

Q1 and the connector P1. If the direct drive video does not work, check U67 to specs. If U67 checks good, check for about 14 MHz on pin 13 of U6. If the signal is bad or not present, check U26 to specs. If pin 13 has a good signal, check for a signal on pin 9 of U101 (CF page 16). If the signal on pin 9 is bad, check U6 to specs. If the signal is good, check U101 to specs. (See flowchart 5-27.)

Problem: No Horizontal Sync (Color)

Symptom Described

Display unreadable due to lack of horizontal sync.



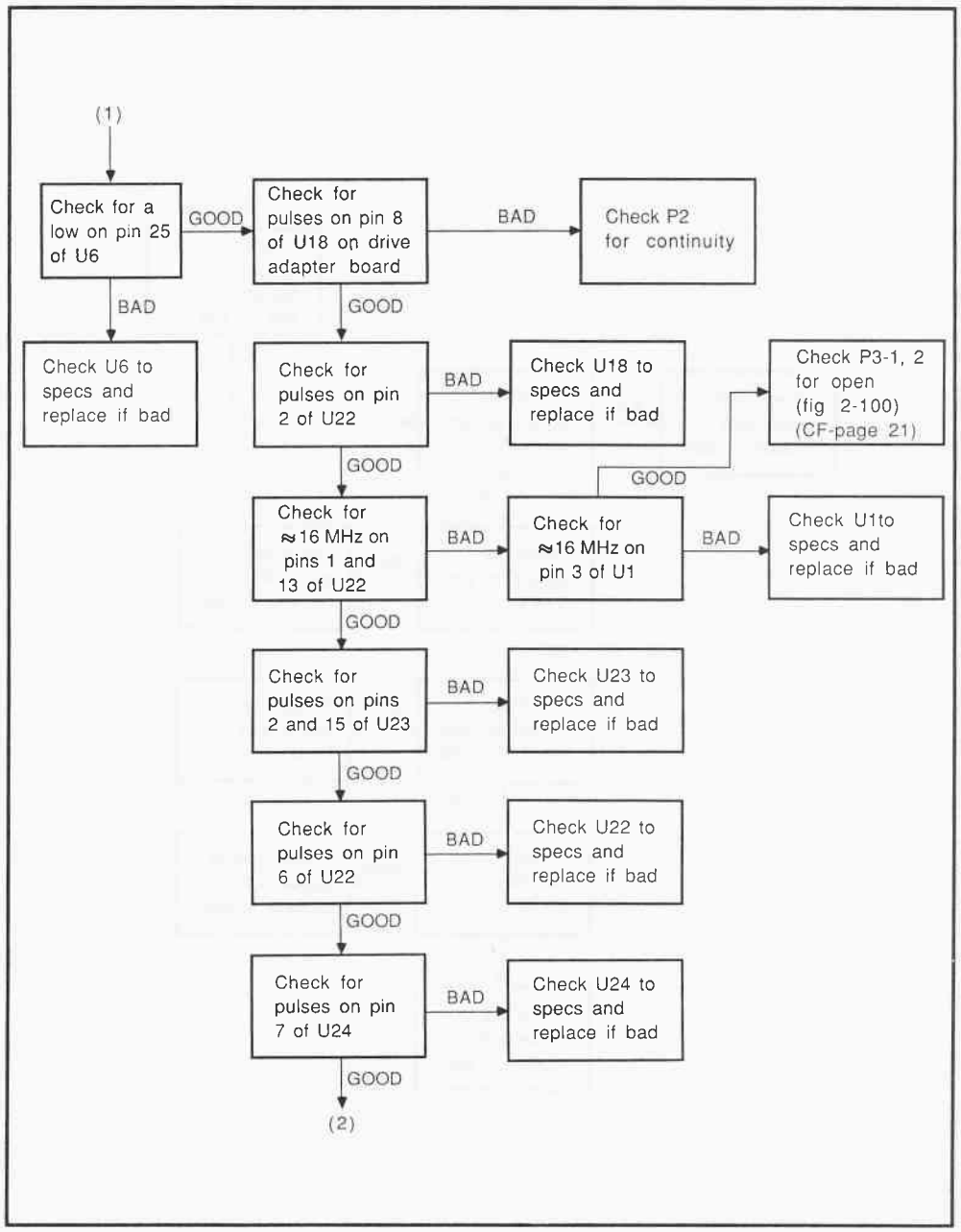
Flowchart 5-11.

Preliminary Checks

1. Check the cable for continuity and proper mating.
2. Clean the adapter card edge connector pins.
3. Try a different known-good display monitor.
4. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-94, 2-95, and CF pages 15, 16, and 17 apply)

Check for a signal on pin 7 of U21. If no signal is found, check U21 to specs. If U21 is good, check U38 by substitution. If good signal is found on pin 7 of U21, check for pulses on pin 8 of U42. If pulses are found, check U67 to specs. If no pulses are found on pin 8, check U64 to specs. If U64 is good, check U42 to specs. (See flowchart 5-28.)



Flowchart 5-11. "cont."

Problem: No Vertical Sync (Color)

Symptom Described

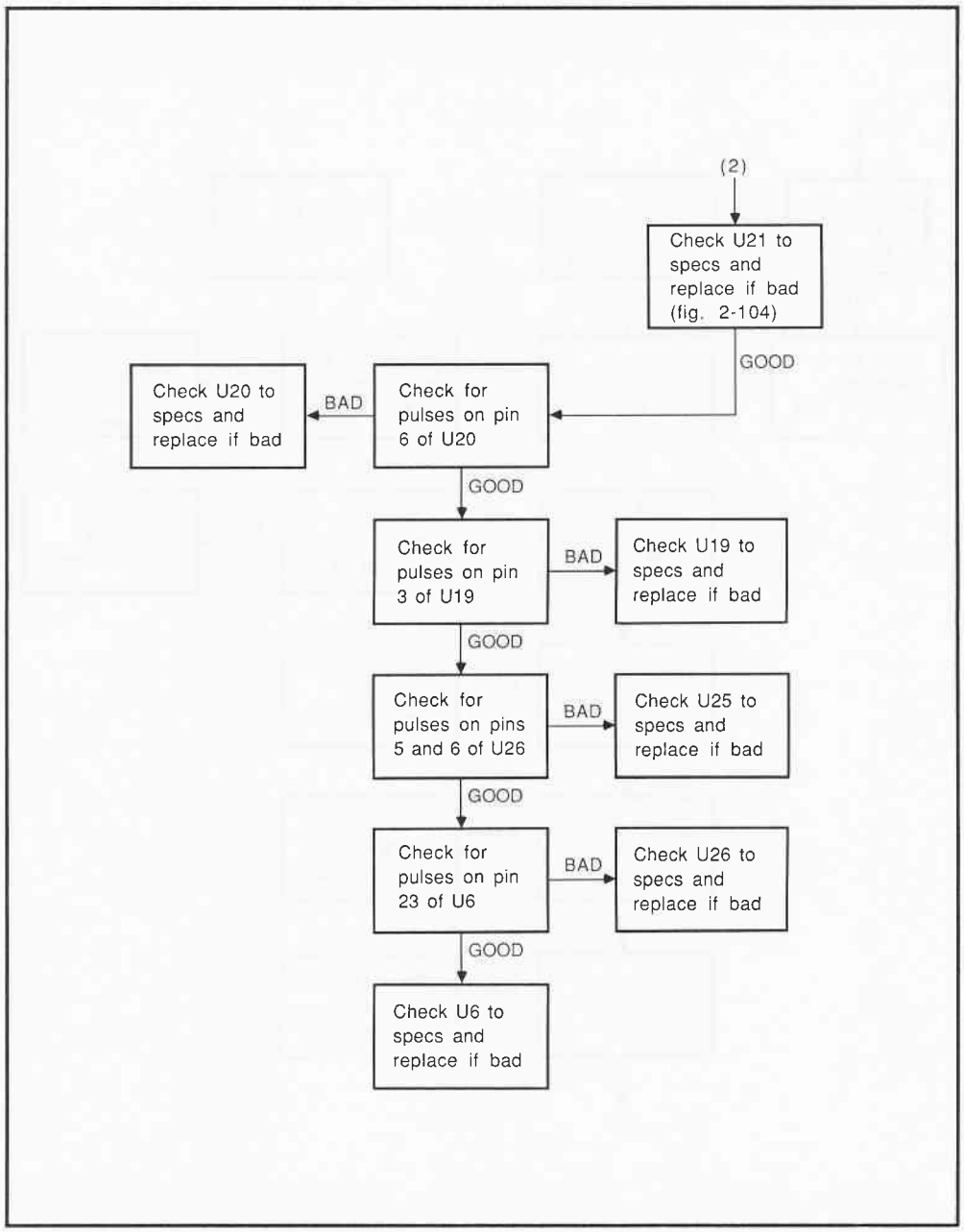
Display continues to roll due to lack of proper vertical sync signal.

Preliminary Checks

1. Check the cable for continuity and proper mating.
2. Clean the adapter card edge connector pins.
3. Try a different known-good display monitor.
4. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-94, 2-95, and CF pages 15, 16, and 17 apply)

Check for signal on pin 11 of U21 (CF page 15). If no signal is present, check U38 by substitution.



Flowchart 5-11. "cont."

If signal is present, check for signal on pin 9 of U67 (CF page 17). If a signal is noted on pin 9, check J2-8 for an open or bad cable. If pin 9 of U67 had signal, check for signal on pin 1 of U63. If pin 1 has signal, check for pulses on pin 9 of U63. If pulses are present, check for pulses on pin 8 of U41. If pulses are found, check U67 to specs. If no pulses are found on pin 8 of U41, check U41 and U63 to specs. If pin 9 of U63 doesn't have pulses check U42 and U64 to specs. If no signal was found on pin 1 of U63, check U21 to specs. (See flowchart 5-29.)

Problem: No Text, Graphics Works (Color)

Symptom Described

When running a graphic program, the display is fine, but when running in text mode, no display or a garbage display is produced.

Preliminary Checks

1. Clean the adapter card edge connector pins.
2. Try a different known-good monitor.

3. Verify that the system is configured properly (switches on system board correctly set).
4. Refer to appropriate section in Chapter 4.

Classical Approach (Fig. 2-93 and CF page 2 apply)

Check U33 by substitution. If problem persists, check U32 by substitution. If this doesn't correct problem, check for pulses on pin 9 of U23. If pulses are present, check U23 to specs and replace if bad. If no pulses are found on pin 9 of U23, check for pulses on pin 4 of U14. If no pulses are found, check for a high on pin 12 of U13. If pin 12 is high, check U13 to specs. If pin 12 is not high, check U14 to specs.

If pulses are found on pin 4 of U14, check for pulses on pin 5 of U14. If pulses are present, check U14 to specs. If no pulses are found on pin 5 of U14, check U49 to specs and replace if bad. (See flowchart 5-30.)

Problem: No Graphics, Text Works

Symptom Described

When trying to run a graphic program, garbage is displayed, or no display is produced, but when running a text program, the display looks fine.

Preliminary Checks

1. Verify system configuration. Check switches on system board.
2. Clean the color graphics adapter card edge connector.
3. Try a different monitor.
4. Refer to appropriate section in Chapter 4.

Classical Approach (Fig. 2-93 and CF page 2 apply)

Check for a high on pin 13 of U23. If pin 13 is not high, check U40 to specs. If pin 13 is high, check U23 to specs. (See flowchart 5-31.)

Problem: Bad Characters (Color Graphics Adapter)

Symptom Described

Garbage on screen, illegal characters, strange shapes in both graphic and text mode.

Preliminary Checks

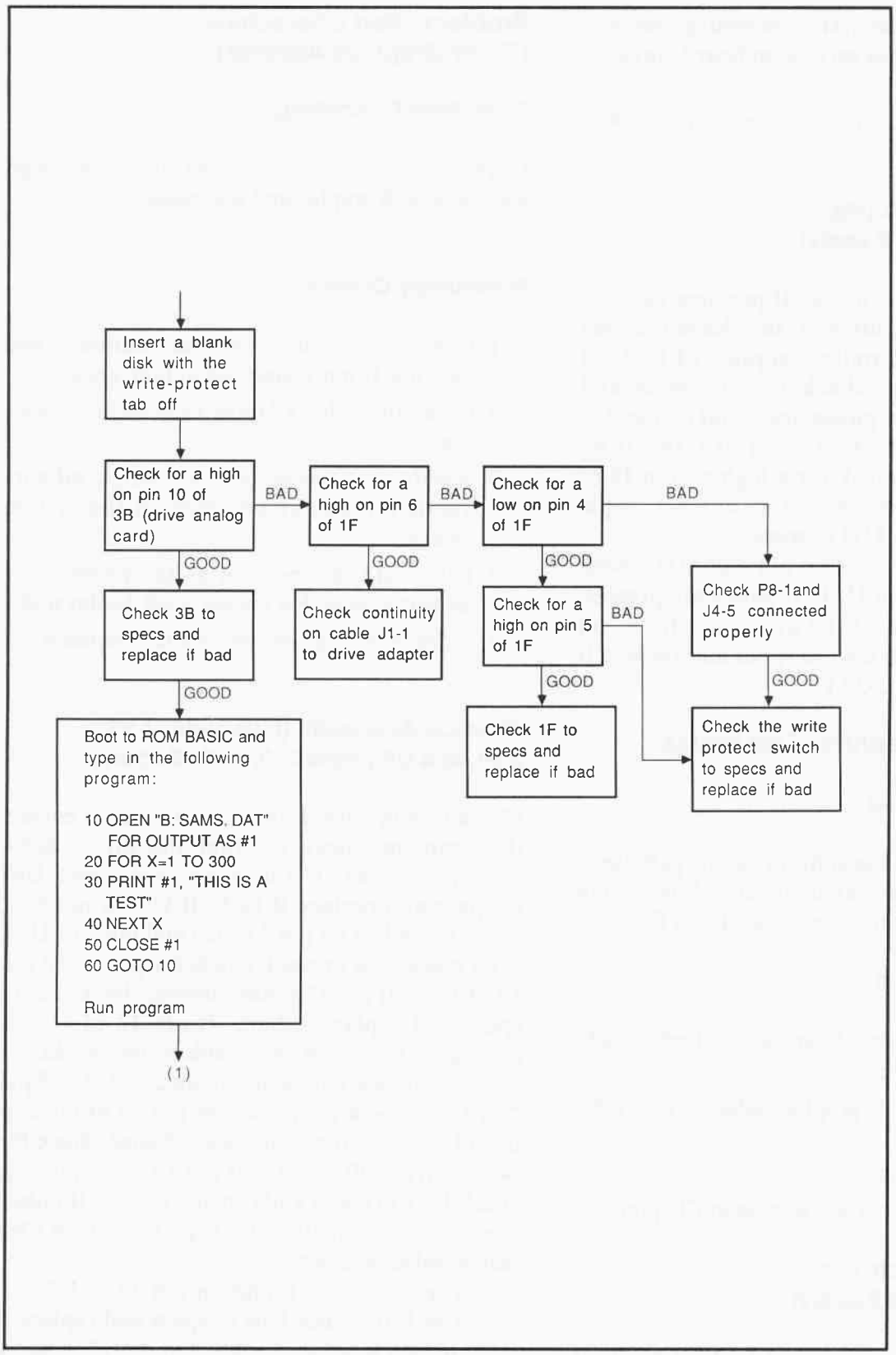
1. Check the monitor cable for continuity and that it is firmly connected at both ends.
2. Clean the color adapter card edge connector.
3. Verify that the system is configured correctly (check the switches on the system board).
4. Eliminate the monitor as the problem by substitution with a known good display unit.
5. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-86, 2-93, 2-94, and CF pages 2, 3, and 15 apply)

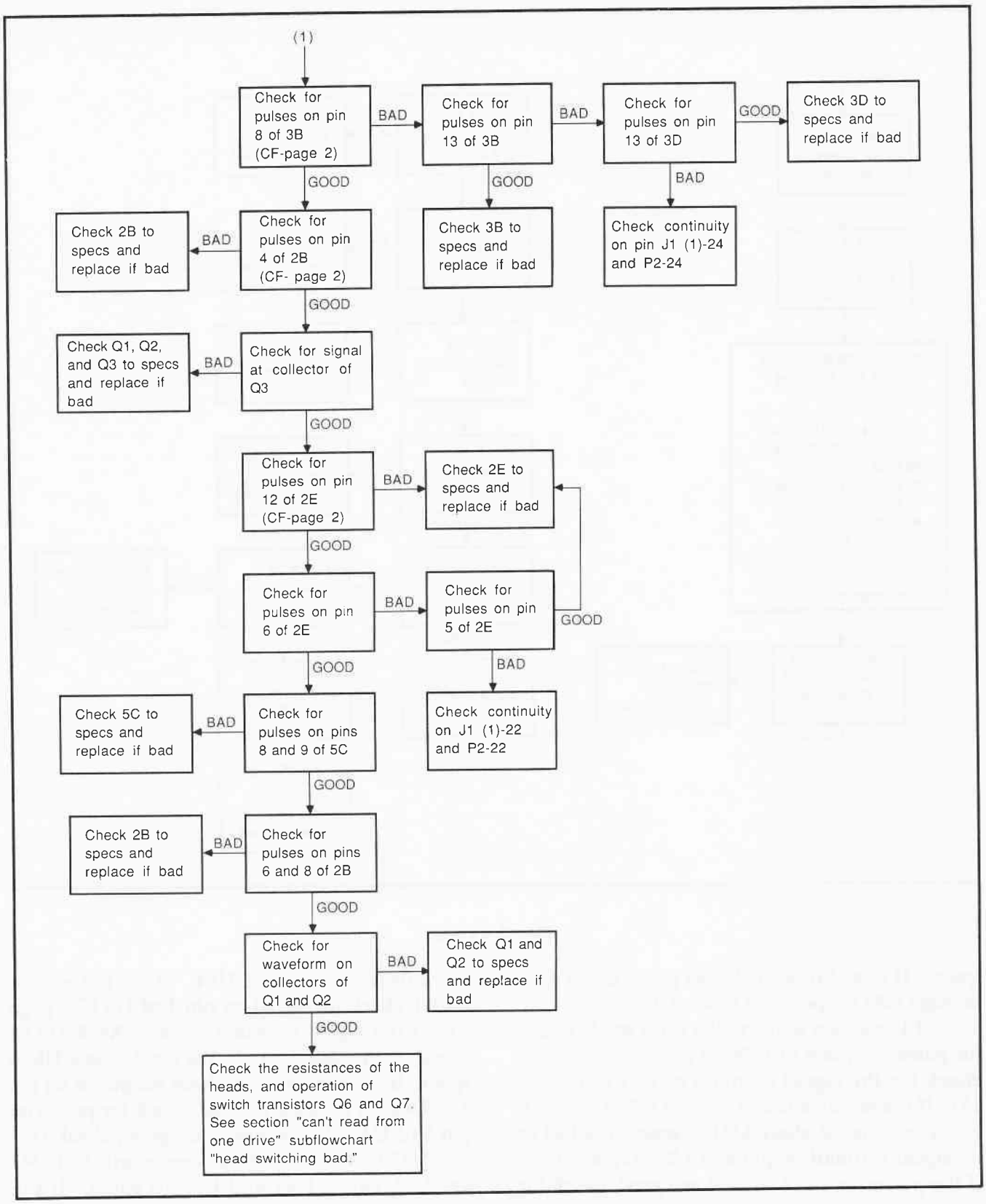
Check U38 by substitution. If that doesn't correct the problem, check whether the direct drive video port works. If it does not work, check U67 to specs and replace if bad. If U67 is not bad, check for pulses on pin 2 of U9 and pin 2 of U10. If no pulses are present, check for pulses on pin 12 of U23. If pin 12 is not pulsing, check U21 to specs and replace if bad. If pin 12 of U23 is pulsing, go to section "No graphics, text works."

If pulses are present on pin 2 of U9 and pin 2 of U10, check for pulses on pin 14 of U9 and pin 14 of U10. If no pulses are found, check for pulses on pin 10 of U23. If pin 10 is not pulsing, check U21 to specs and replace if bad. If pulses are found on pin 10 of U23, go to section "No text, graphics works."

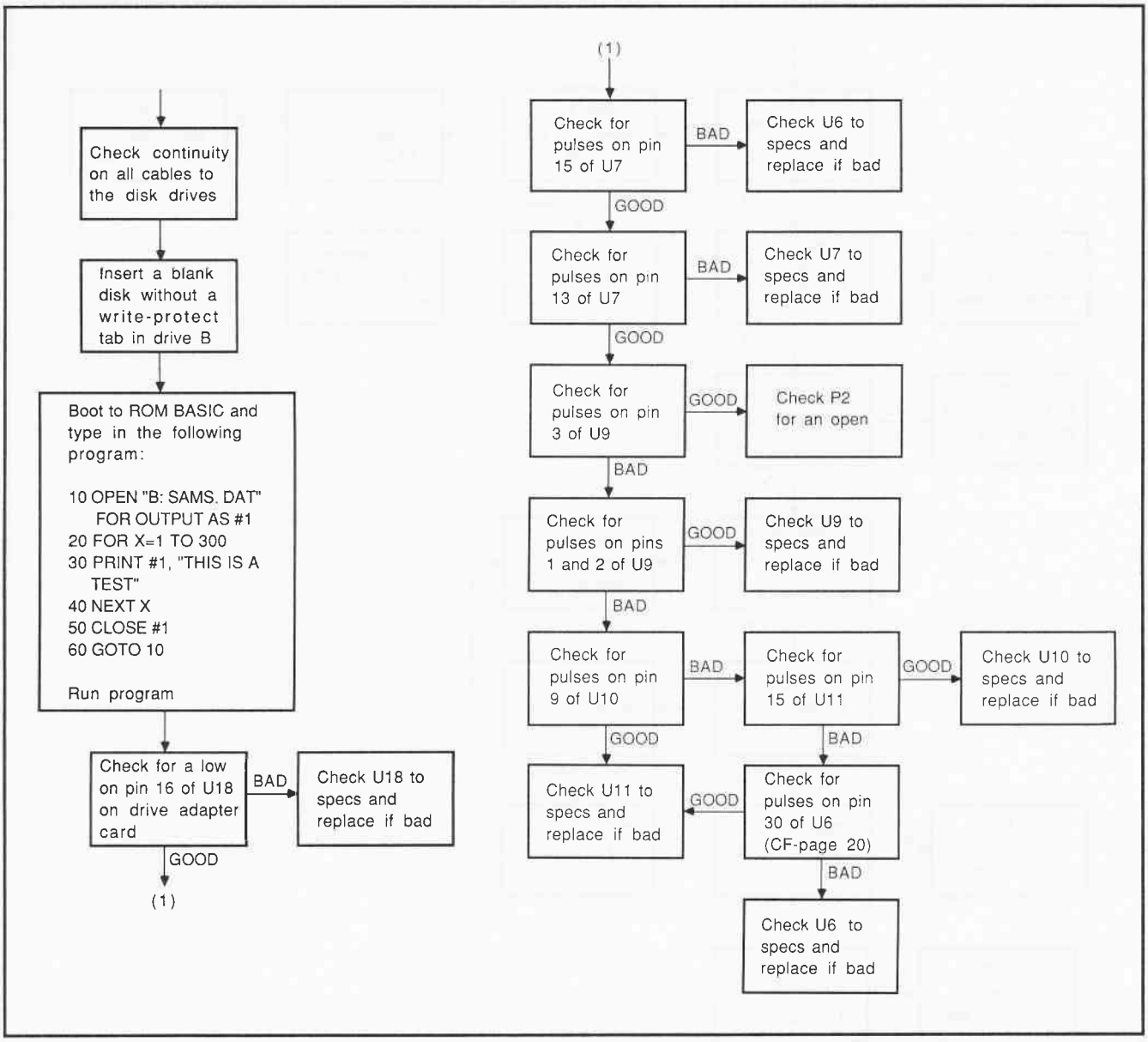
If pulses were found on pin 14 of U9 and pin 14 of U10, check U66 to specs and replace if bad. If U66 is good, check U9 and U10 to specs and replace if bad. If U9 and U10 check good, check U36 to specs. If U36 is good, check U58 to



Flowchart 5-12.



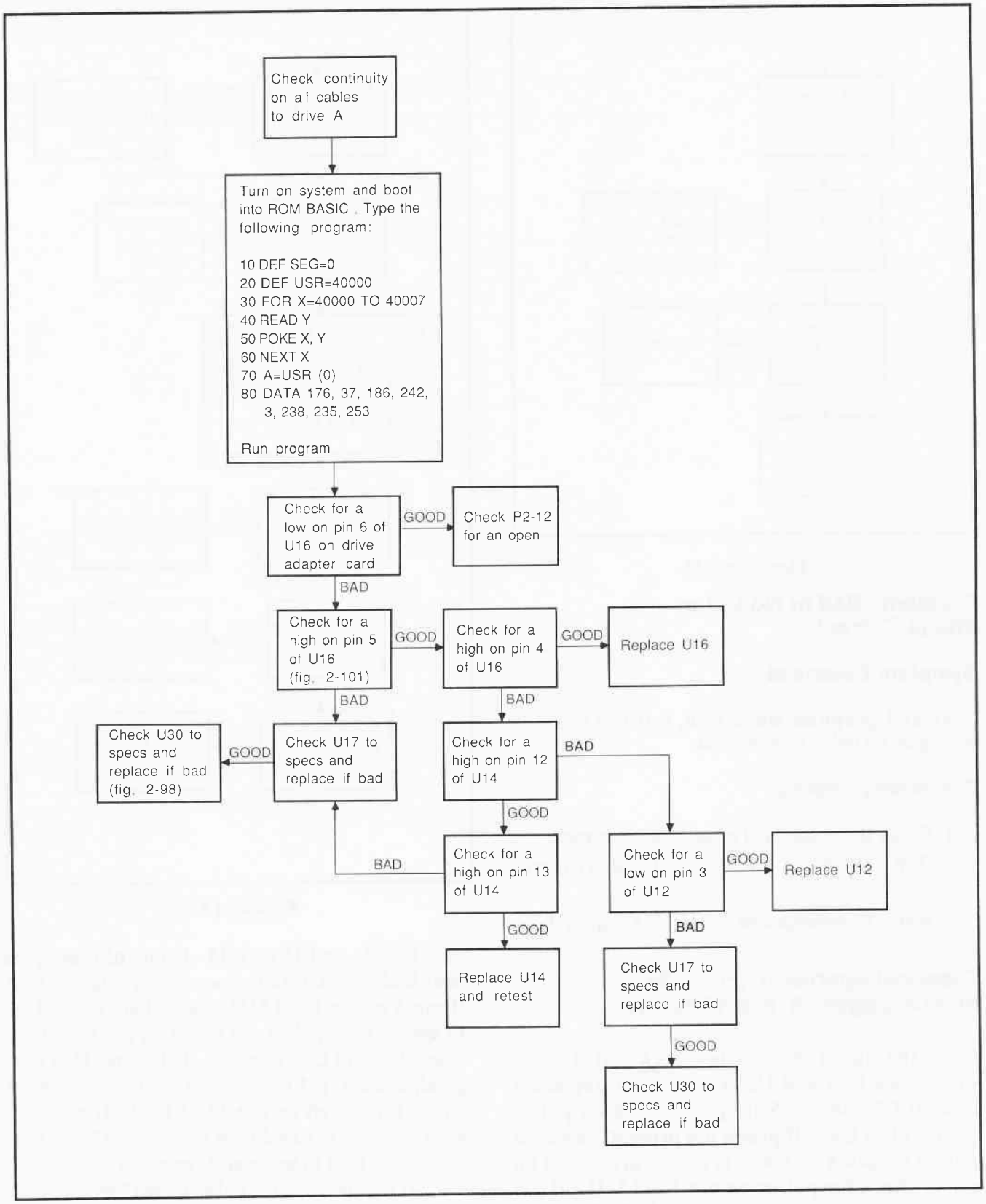
Flowchart 5-12. "cont."



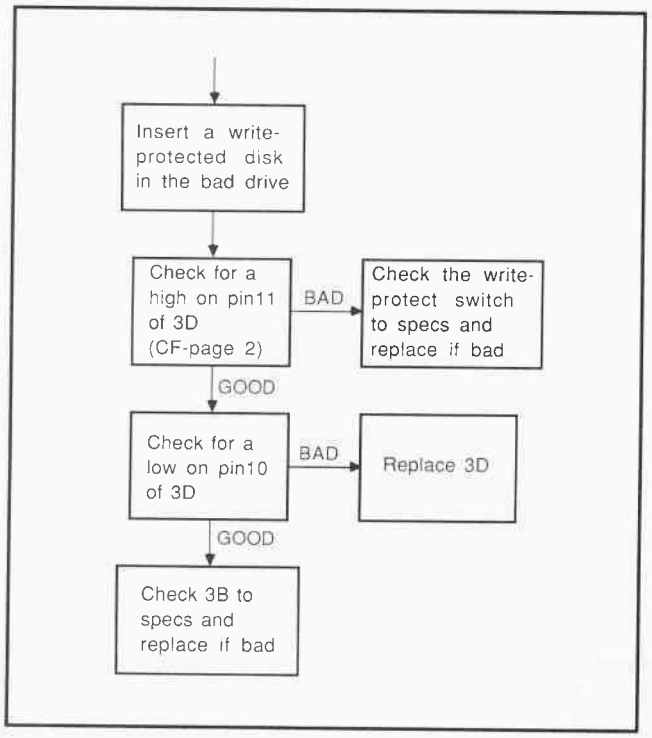
Flowchart 5-13.

specs. If this chip also checks good, check U50 through U57 to specs and replace if bad.

If the direct drive video does work, check for pulses on pin 4 of U24. If pulses are missing, check for the signal on pin 6 of U21 (CF page 15). If the signal is found, check U65 to specs. If U65 checks good, check U21 to specs. If a bad or no signal is found on pin 6 of U21, replace U38. If the pulses on pin 4 of U24 are good, check for pulses on pin 2 of U24. If pulses are present, check U24 to specs and replace if bad. If the pulses on pin 2 of U24 are bad or missing, check for pulses on pin 8 of U64. If no pulses are found, check for signal on pin 1 of U6 (CF page 3). If the signal on pin 1 is bad, check U4 to specs. If the signal on pin 1 is good, check U6 to specs. If U6 checks good, check for pulses on pin 4 of U42. If pulses are found, check for pulses on pin 5 of U42. If the pulses are good, check U20 and U42 to specs. If the pulses on pin 4 of U42 are bad, check U41 and U63 to specs. If the pulses on pin 5 of U42 are bad, check U64 to specs. If U64 checks good, check U4 and U6 to specs.



Flowchart 5-14.



Flowchart 5-15.

Problem: Bad or No Color–Image Correct

Symptom Described

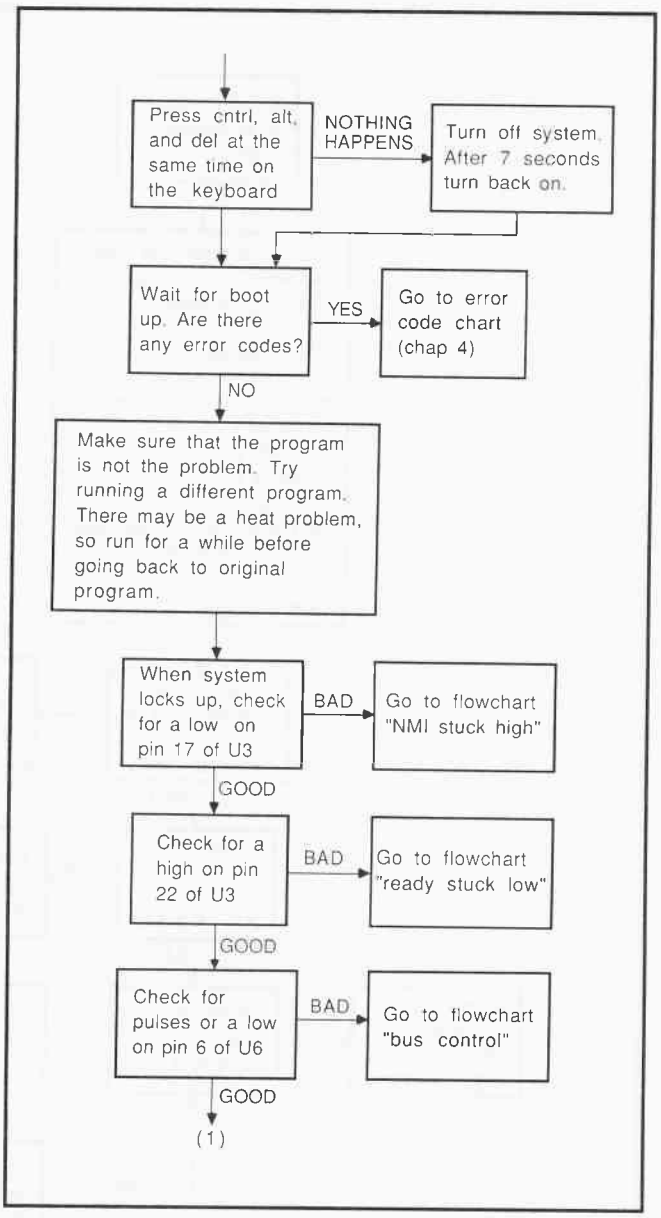
Text and graphics work fine, but the color is wrong or no color is produced.

Preliminary Checks

1. Clean the color card edge connector pins.
2. Try a different known-good display monitor.
3. Refer to appropriate section in Chapter 4.

Classical Approach (Fig. 2-94 and CF pages 16 and 17 apply)

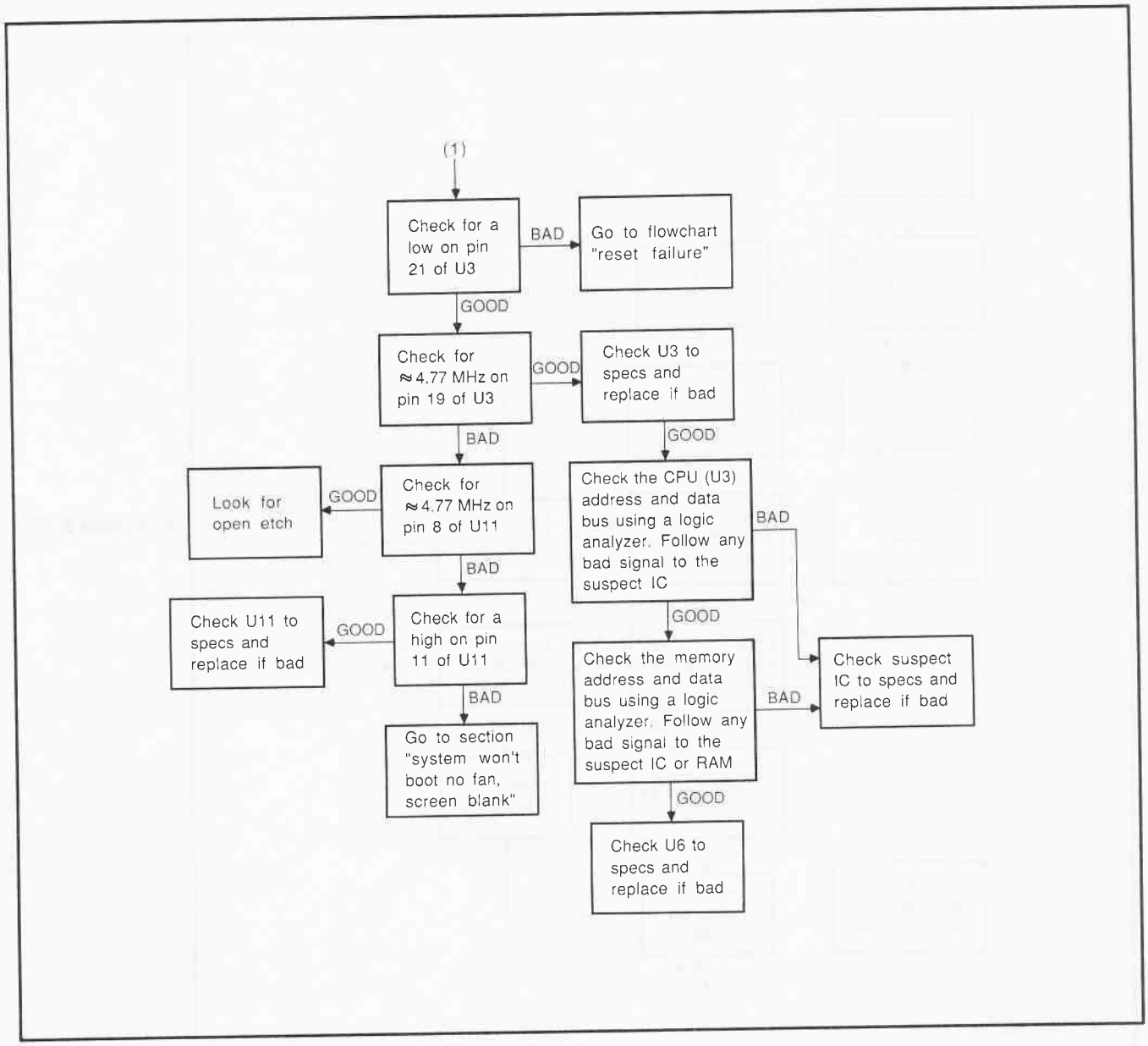
Does the direct drive video work? If it does, then check U24 and U25 to specs and replace if bad. If U24 and U45 are good, check for pulses on pin 11 of U45. If pulses are present, check for pulses on pin 4 of U45. If pulses are found on pin 4, check for pulses on pin 10 of U45. If pin 10 is good, check for pulses on pin 3 of U45. If pulses are found on pin 3, check for pulses on pin 13 of U45. If pin 13 is good, check for pulses on



Flowchart 5-16.

pins 1, 2, 14 and 15 of U45. If one of these pins tests bad, check U44 to specs and replace if bad. If pin 3 or pin 13 of U45 checks bad, check U43 to specs. If no pulses are found on pin 10 of U45, check U9 and U10 to specs. If U9 and U10 tests good, check U101 to specs. If U101 checks good, check for a high on pin 12 of U14. If pin 12 is high, check U65 and U68 to specs. If pin 12 is low, check U20 to specs and replace if bad.

If the direct drive video doesn't work, check U67 to specs and replace if bad. If U67 tests good, check U101, U9, and U10 to specs. (See flowchart 5-33.)



Flowchart 5-16. "cont."

Problem: Cursor Missing or Not Blinking

Symptom Described

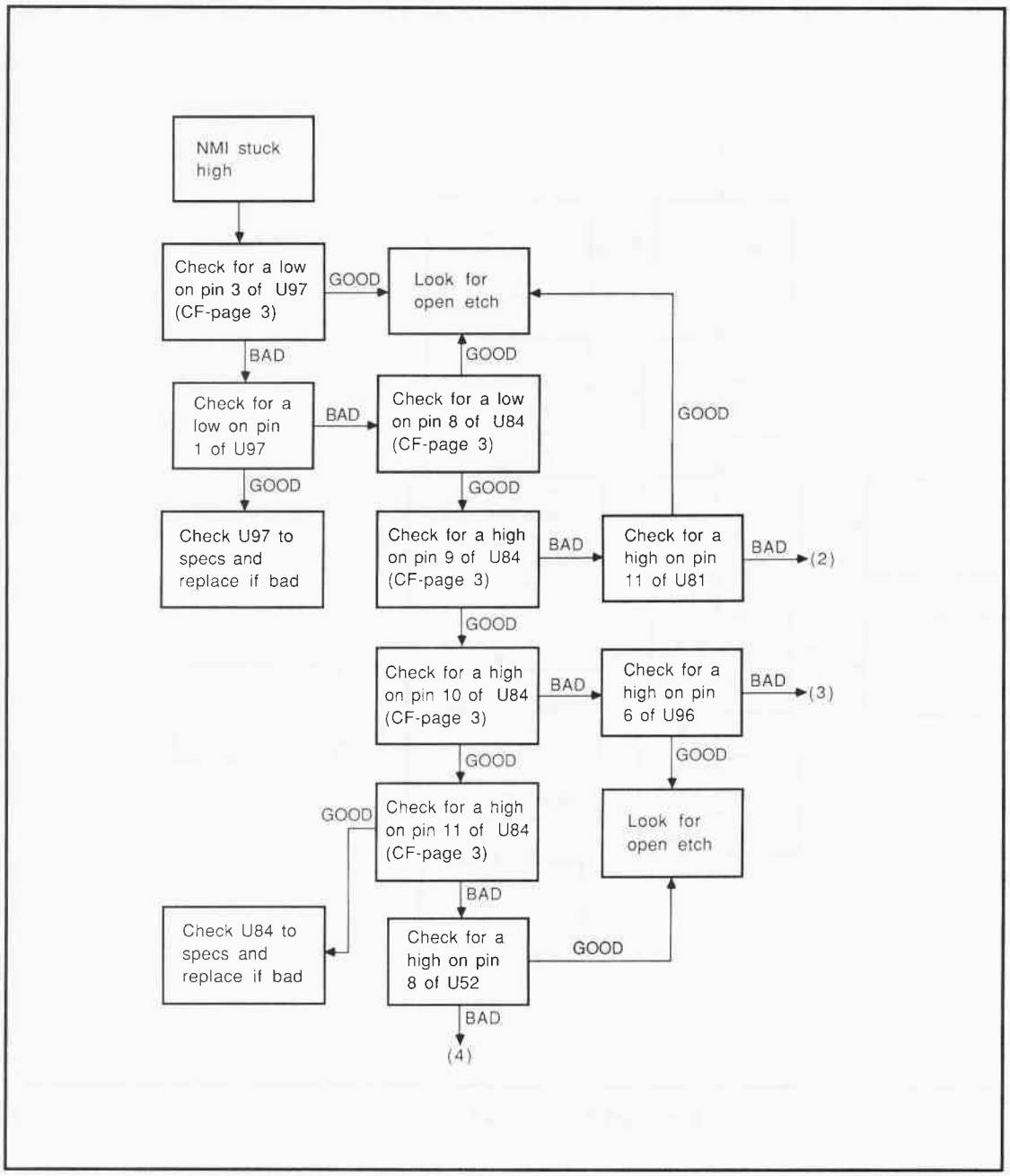
No cursor on display or cursor is present, but not blinking.

Preliminary Checks

1. Clean color card edge connector pins.
2. Try a different, known-good display monitor.
3. Refer to appropriate section in Chapter 4.

Classical Approach (Figs. 2-93, and CF pages 2 and 15 apply)

Check for pulses on pin 2 of U20. If pulses are present, check U21 to specs and replace if bad. If no pulses are found, check U12 to specs and replace if bad. If U12 tests good, check for pulses on pin 19 of U38. If pulses are present on pin 19 of U38, check U20 to specs. If no pulses are found on pin 19, check U38 to specs and replace if bad. (See flowchart 5-34.)



Flowchart 5-17.

4. IBM PC KEYBOARD PROBLEMS

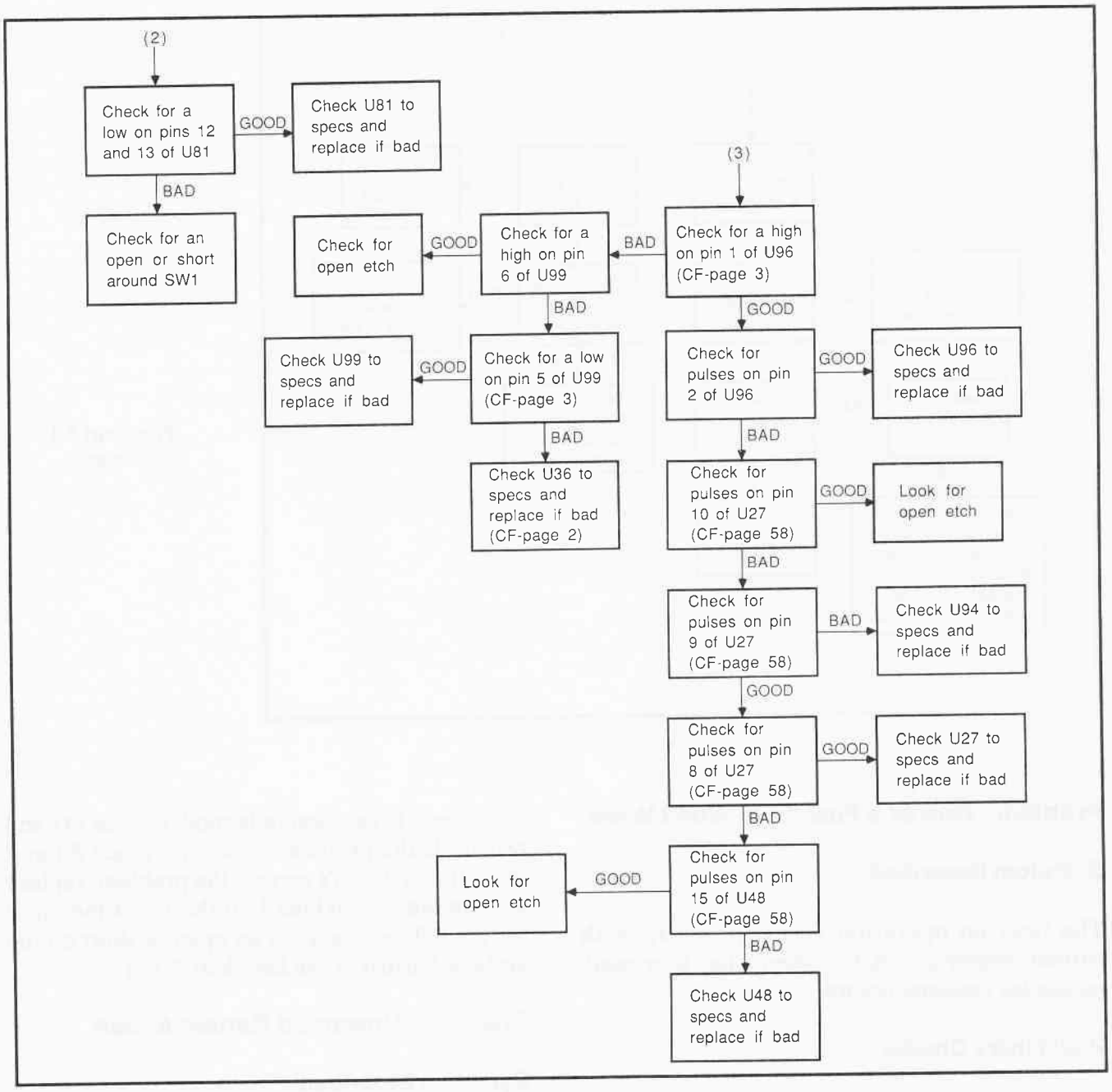
Problem: Keyboard Won't Respond at All or Wrong Character Is Produced

Symptom Described

When any key is pressed, no display response is noted; or when a key is pressed, the wrong character is displayed on the monitor.

Preliminary Checks

1. Check the video cable for continuity and proper mating.
2. Clean the keys with a tuner cleaning spray.
3. Refer to appropriate section in Chapter 4.

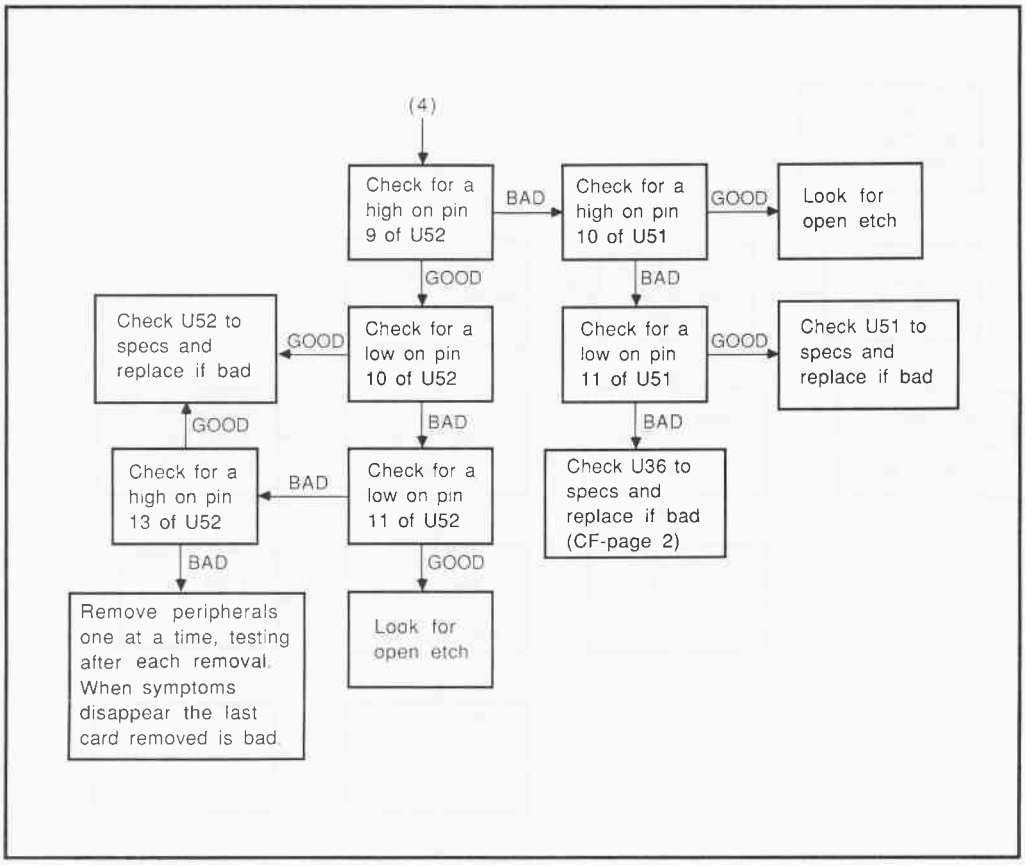


Flowchart 5-17. "cont."

Classical Approach (Figs. 2-59, 2-60, and CF page 2 apply)

While pressing a key, check for a signal on pin 17 of U24 (system board through CF page 2). If the signal is bad, check for proper signal on pin 3 of M2 in the keyboard. If pin 3 is bad, check to see if M1 and M3 are swapping. If pin 3 of M2 is good, check M2 by replacement. If pin 3 is still bad, check Z1 and replace if bad.

If pin 17 of U24 is good, while pressing a key, check for a signal on pin 1 of U24. If a bad or no signal is found, check U82 to specs and replace if bad. If pin 1 of U24 is good, check U36 to specs and replace if bad. If U36 also tests good, check for a signal on pin 11 of U24. If pin 11 tests bad, check U26 and U80 to specs and replace if bad. If pin 11 tests good, check U24 by replacement. (See flowchart 5-35.)



Flowchart 5-17. "cont."

Problem: One or a Few Keys Won't Work

Symptom Described

The boot-up operation occurs properly, with correct display action, but when a key is pressed, no display response occurs.

Preliminary Checks

1. Verify that the keyboard-system board cable has continuity and is properly connected.
2. Clean the keys with a tuner cleaner spray.
3. Refer to appropriate section in Chapter 4.

Classical Approach (Fig. 2-58, 2-59, and CF page 2 apply)

Check the continuity of the key that is bad. Measuring across the key, verify an open with the key unpressed, and shorted when the key is pressed. If key closure is good, replace M1 and retest. If the problem persists, replace Z1 and test. If this doesn't correct the problem, replace U24 on the system board. If this is not the cause for the failure, check for an open or short on the keyboard matrix. (See flowchart 5-36.)

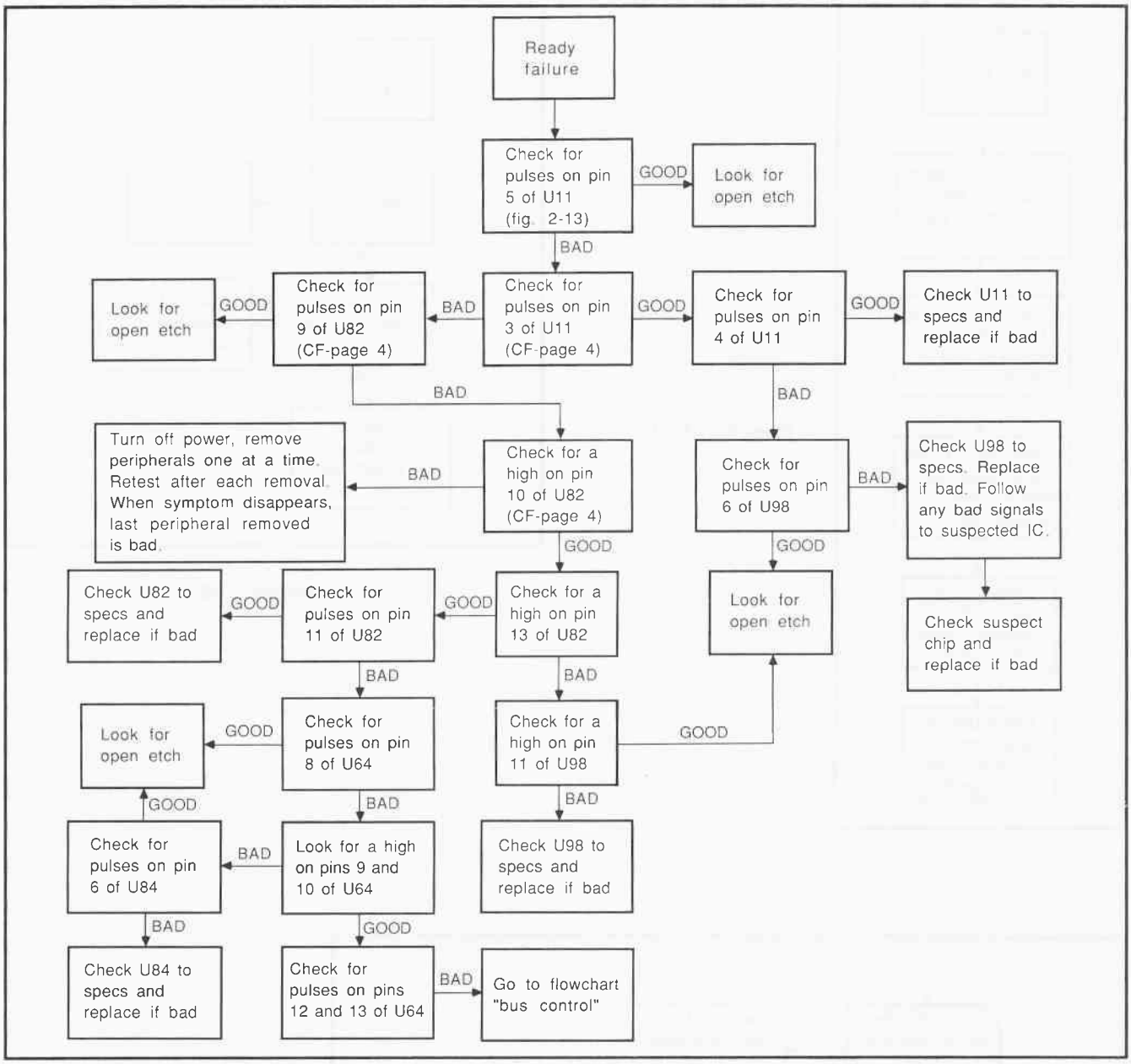
Problem: Unwanted Repeat Action

Symptom Described

When a key is pressed, more than one image of the same character is displayed on the screen. This occurs even when not holding the key down for an intended repeat action.

Preliminary Checks

1. Check the cable for continuity and proper mating.
2. Clean the keyboard keys with a tuner spray.



Flowchart 5-18.

3. If only one key fails, replace the key.
4. Refer to appropriate section in Chapter 4.

Classical Approach (Fig. 2-58 and CF page 2 apply)

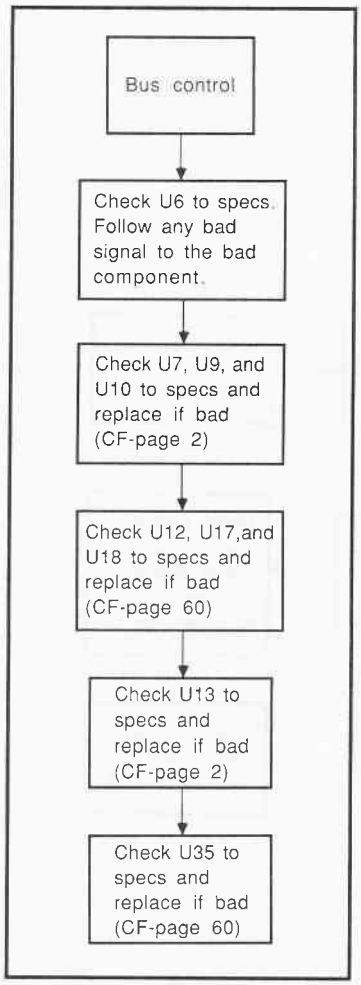
Replace M1 in the keyboard chassis. If this does not correct the problem, check Z1 to specs and replace if bad. (See flowchart 5-37.)

5. IBM PC INPUT/ OUTPUT PROBLEMS

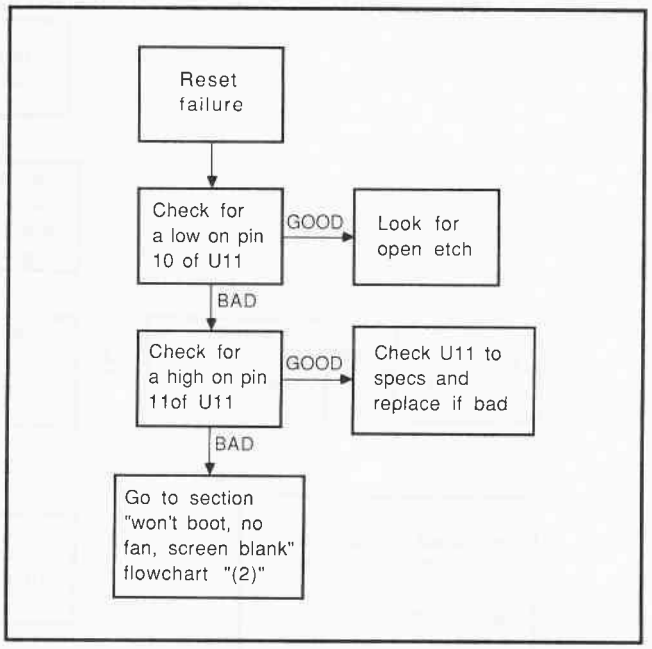
Problem: Cassette—Can't Write Data to Tape

Symptom Described

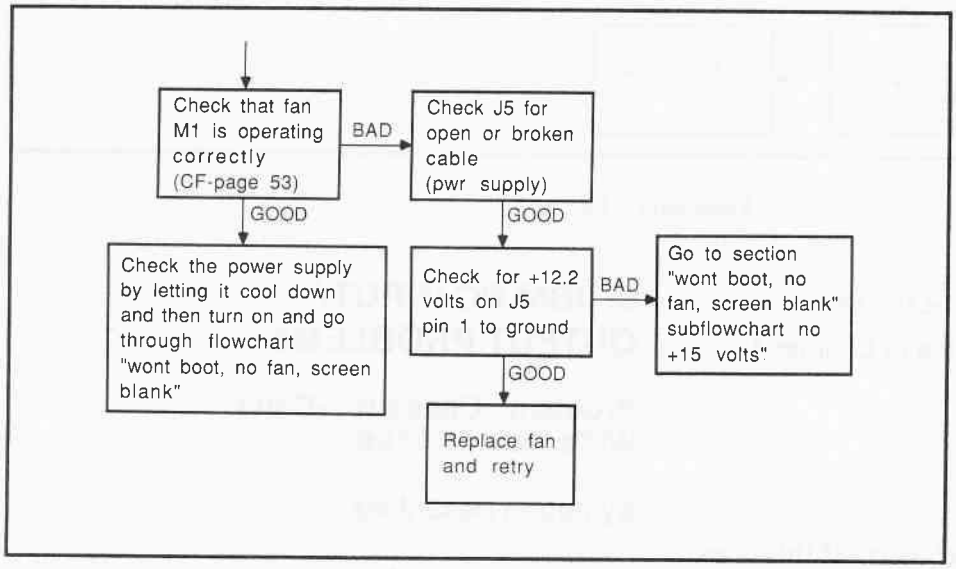
When trying to write to the tape, an error occurs or nothing is written.



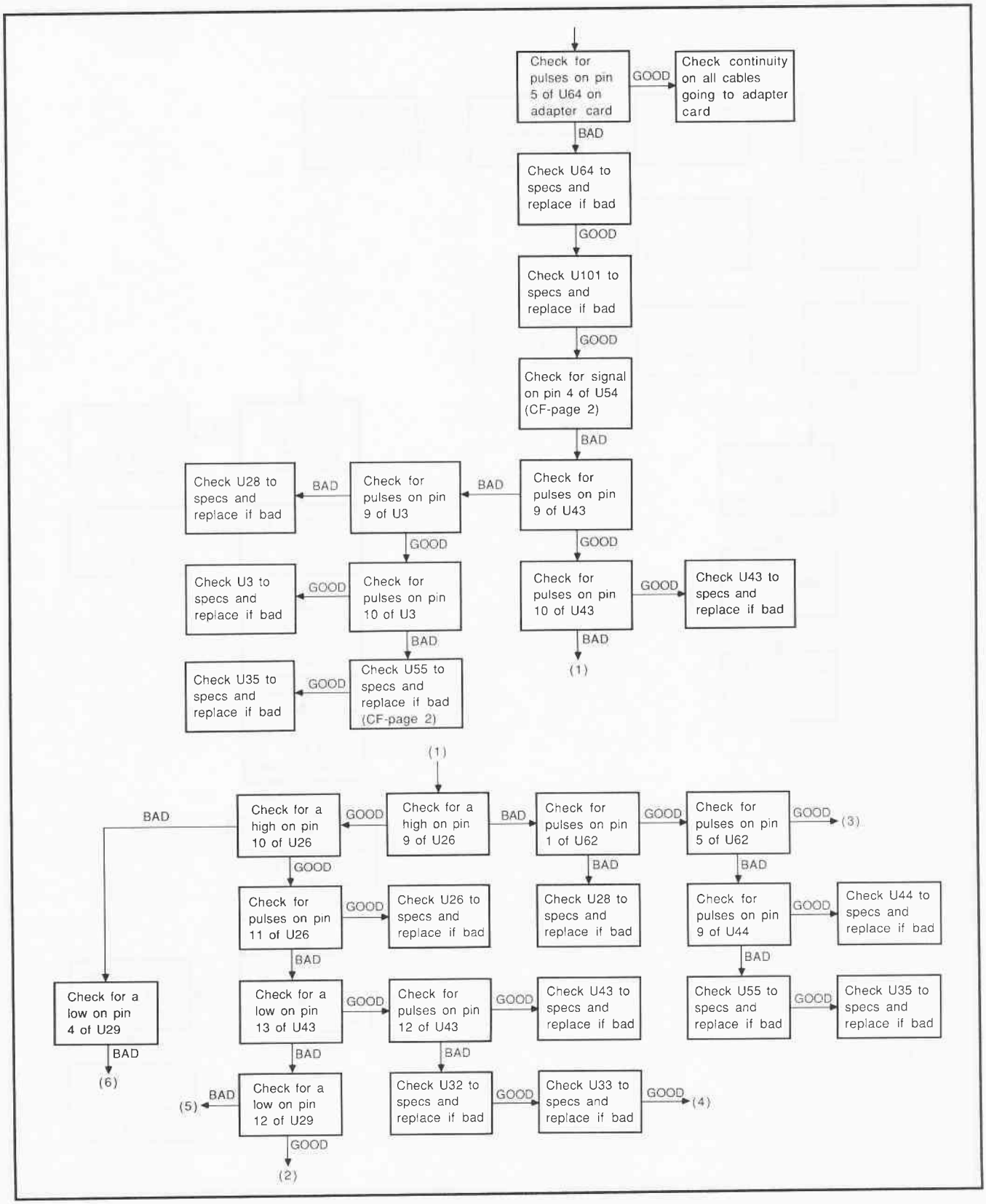
Flowchart 5-19.



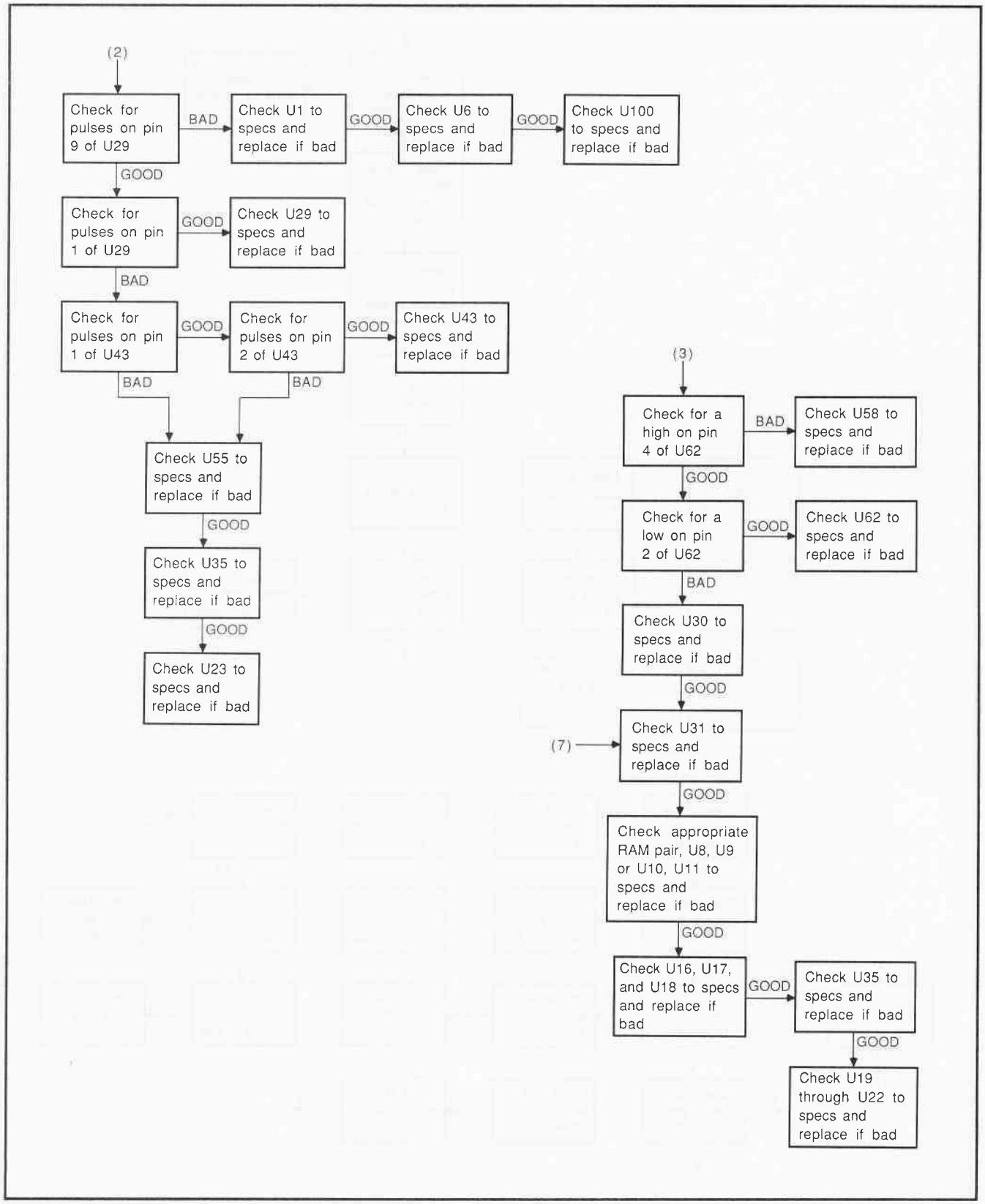
Flowchart 5-20.



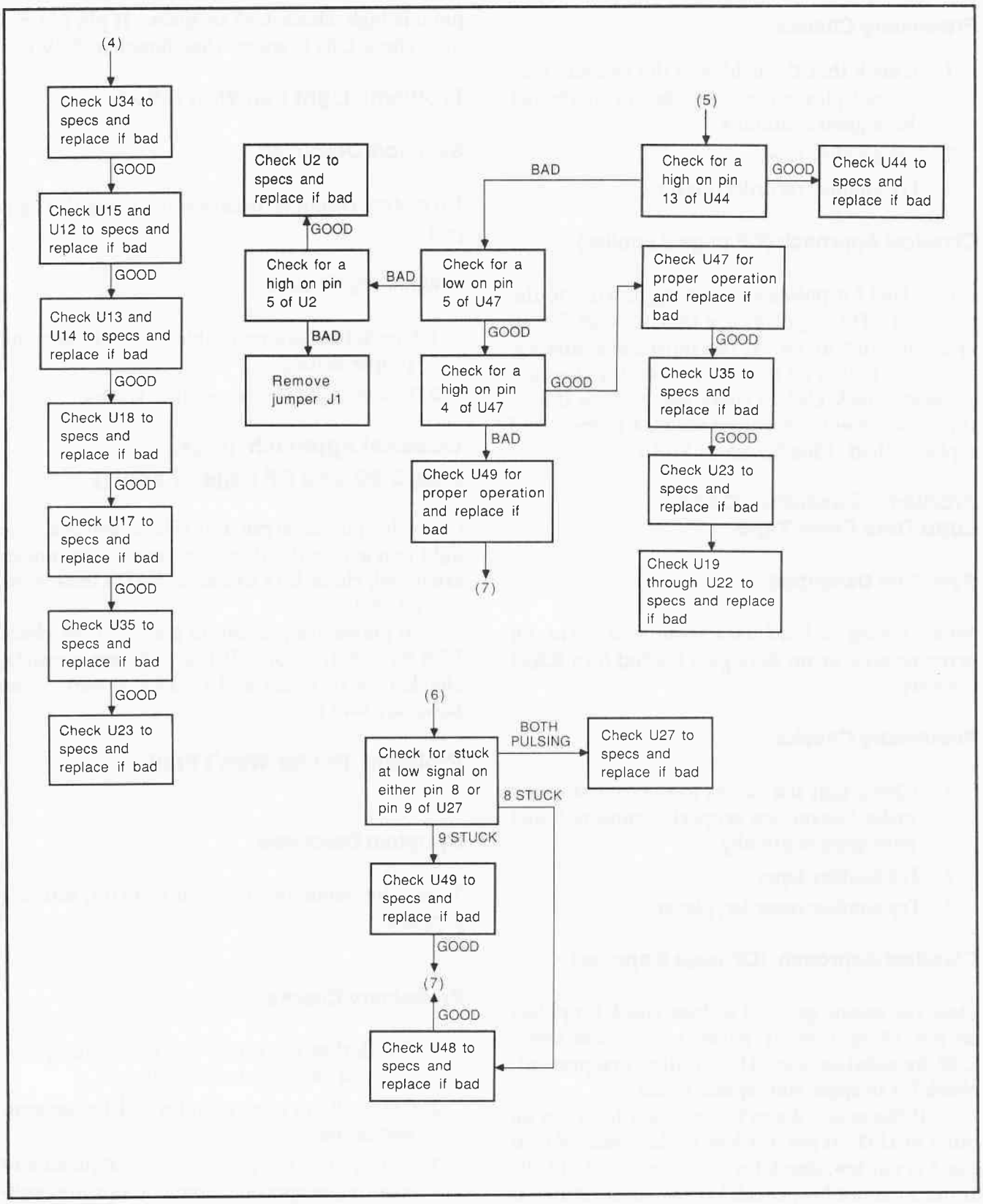
Flowchart 5-21.



Flowchart 5-22.



Flowchart 5-22. "cont."



Flowchart 5-22. "cont."

Preliminary Checks

1. Check that the cables to the cassette recorder/ player are connected properly and have good continuity.
2. Try another tape.
3. Try another recorder/player.

Classical Approach (CF page 2 applies)

Check U63 for pulses when trying to write to the recorder. If the pulses are there, check for an open or short at P4. If the pulses are missing, check for pulses on pin 17 of U34. If pulses are present, check U63 to specs and replace if bad. If pulses are not present, check U34 to specs and replace if bad. (See flowchart 5-38.)

Problem: Cassette—Can't Load Data From Tape

Symptom Described

When trying to load data from a cassette, an error occurs, or no data gets loaded into RAM memory.

Preliminary Checks

1. Check that the cables to the cassette recorder/player are properly connected and have good continuity.
2. Try another tape.
3. Try another recorder/player.

Classical Approach (CF page 2 applies)

Does the motor spin? If it does, check for pulses on pin 13 of U36. If pulses are present, check U36 by substitution. If no pulses are present, check U1 to specs and replace if bad.

If the motor doesn't spin, check for a low on pin 3 of U95. If pin 3 is low, replace relay K1. If pin 3 is not low, check for a low on pin 21 of U36. If pin 21 is not low, check U36 by substitution. If pin 21 is low, check for a high on pin 6 of U63. If pin 6 is high, check U95 to specs. If pin 6 is not high, check U63 to specs. (See flowchart 5-39.)

Problem: Light Pen Won't Work

Symptom Described

No system response occurs when using the light pen

Preliminary Checks

1. Check the light pen cable for continuity and proper mating.
2. Test the light pen on another system.

Classical Approach (Figs. 2-85, 2-90, and CF page 17 apply)

Check for pulses on pin 3 of U38 as you draw the light pen across the display screen. If no pulses are noted, check U11 to specs. If U11 tests good, check U29 to specs.

If pulses are present on pin 3 of U38, check U38 by substitution. If the problem persists, check U24 to specs and replace if bad. (See flowchart 5-40.)

Problem: Printer Won't Print

Symptom Described

When the command is given to print, nothing happens.

Preliminary Checks

1. Check that the cables are connected properly and have good continuity.
2. Clean the printer adapter card edge connector pins.
3. Verify that the printer is configured and working properly. Conduct a printer self-test.

Classical Approach (CF pages 2, 7, and 54 apply if using printer adapter card; CF pages 2 and 13 apply if using monochrome monitor/printer adapter card)

The following description assumes a monolithic printer adapter card. For the monochrome monitor/ printer adapter card, map the ICs as follows:

Printer Adapter	Monochrome Monitor/ Printer Adapter
U1	U23
U6	U61
U7	U39
U8	U38

While running a program that continuously outputs data to the printer, check for signal on pin 8 of U5. If the signal is not present, replace U5. If a signal is present, check for pulses on pin 2 of U8. If the pulses are present, check U8 to specs and replace if bad. If no pulses are present on pin 2, check U7 to specs. If U7 is good, check U6 to specs. If U6 is good, check U1 to specs. (See flowchart 5-41.)

Problem: Printer Locks Up or Prints Garbage

Symptom Described

When beginning or in a print operation, the printer stops or begins printing garbage.

Preliminary Checks

1. Check that the cables are properly connected and have good continuity.
2. Clean the printer adapter card edge connector pins.
3. Verify that the printer is configured properly and operates properly. Conduct a printer self-test.

Classical Approach (CF pages 2, 7, and 54 if using printer adapter card; CF pages 2 and 13 if using monochrome monitor/printer adapter card)

The following description assumes a monolithic printer adapter card. For the monochrome monitor/printer adapter card, map the ICs as follows:

Printer Adapter	Monochrome Monitor/ Printer Adapter
U2	U37
U3	U40
U4	U41

While running a program that continuously outputs to the printer, check U2 to specs and replace if bad. If U2 is good, check U4 and U3 to specs. Replace if bad. If the problem persists, go to section "Printer won't print." (See flowchart 5-42.)

Problem: Speaker Won't Work

Symptom Described

No noise from speaker.

Preliminary Checks

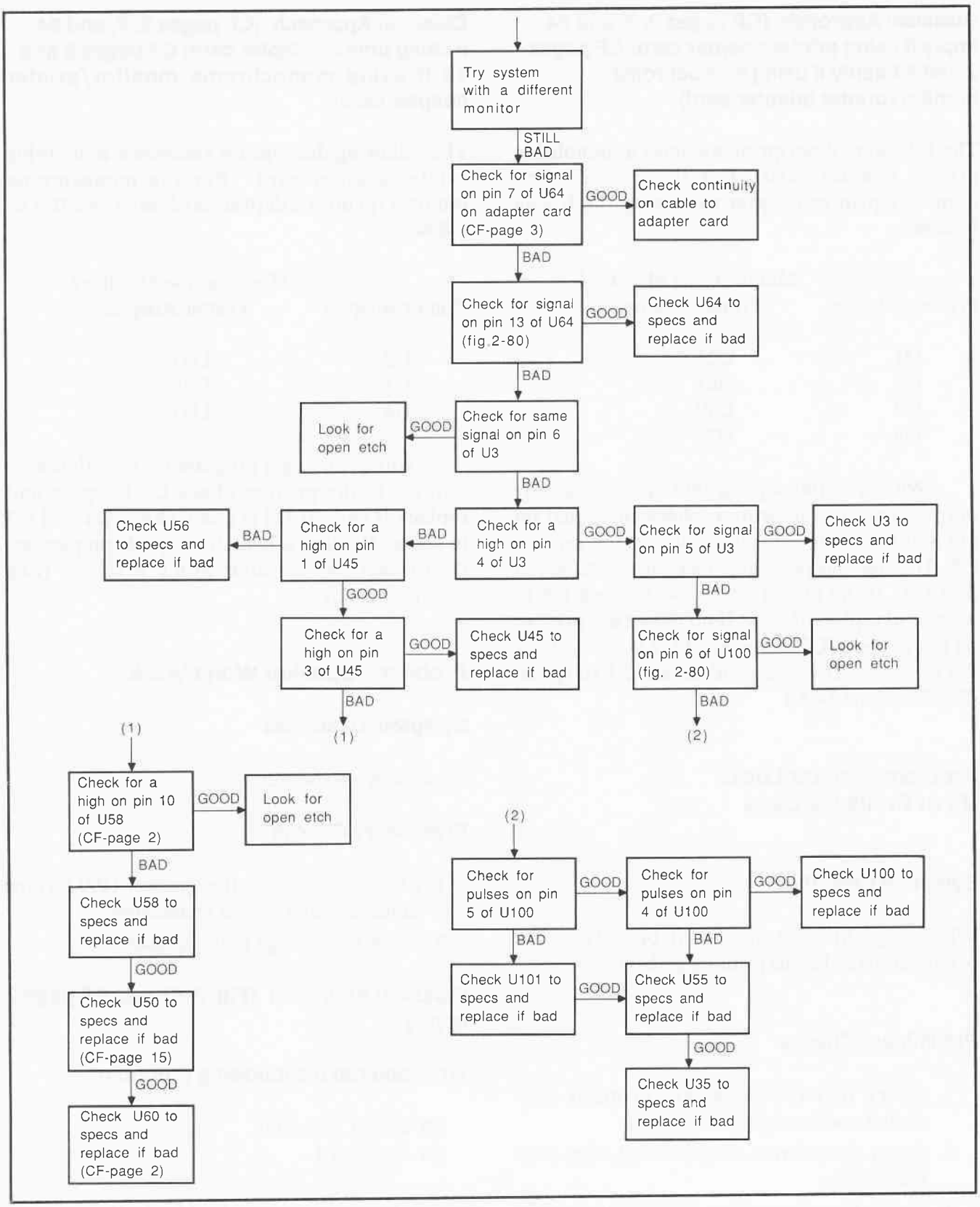
1. Check the cable to the speaker (P/J3A) for continuity and proper connection.
2. Check for damage to the speaker.

Classical Approach (Fig. 2-57 and CF page 2 apply)

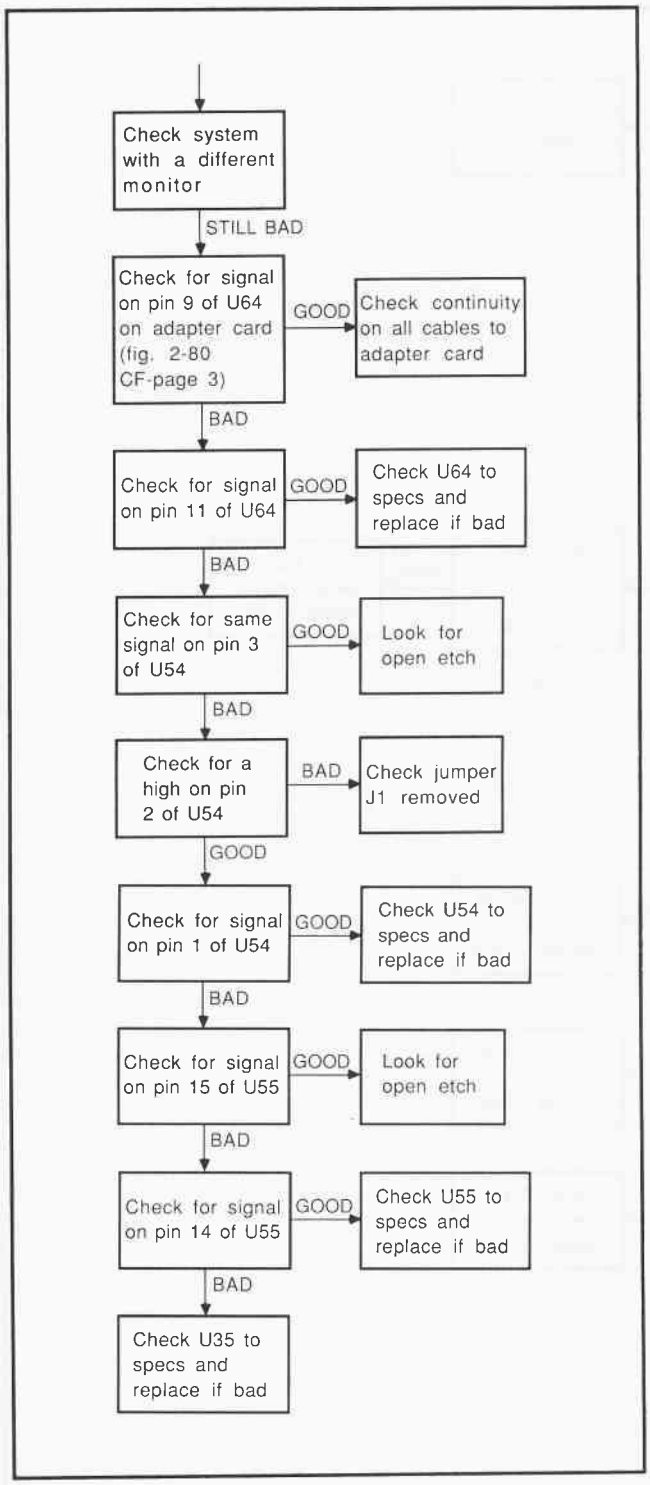
Write and run the following program:

```
10 SOUND 200,200
20 GO TO 10
```

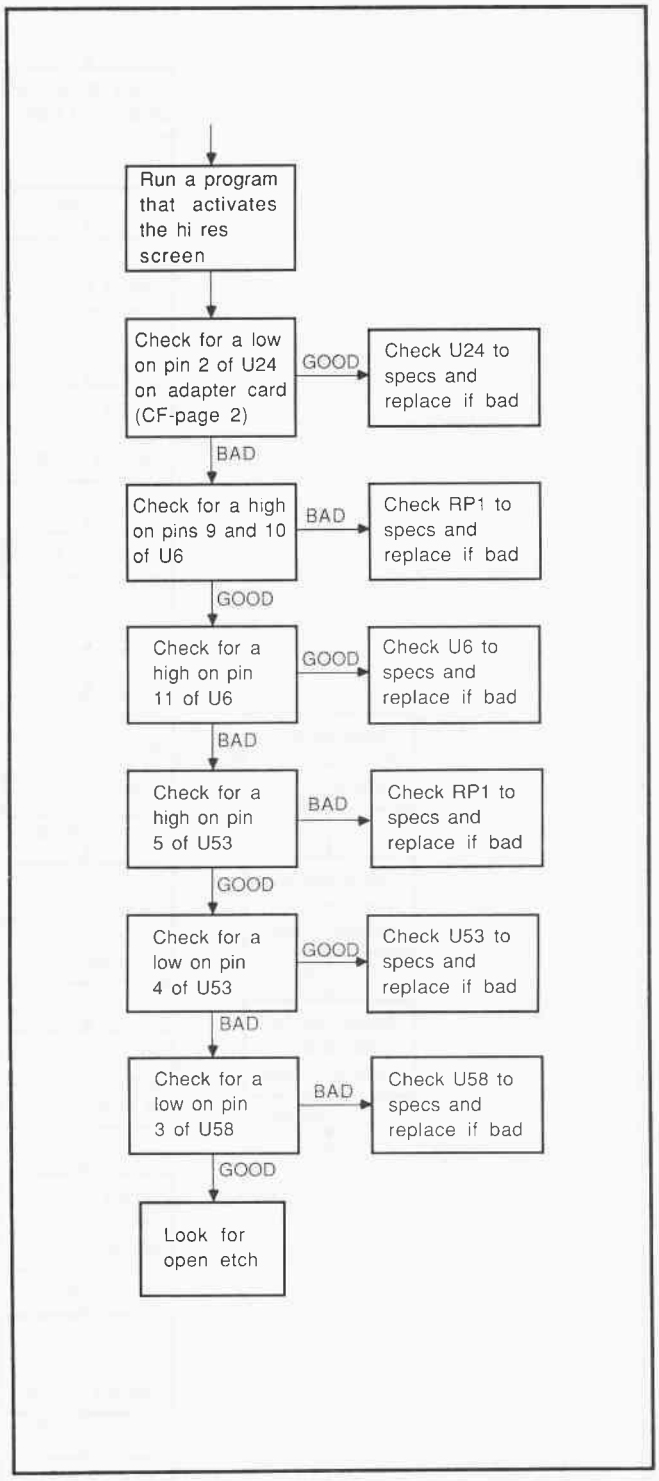
Check for pulses on pin 6 of U95. If pulses are present, replace the speaker. If no pulses are present on pin 6, check for a signal (CF page 57)



Flowchart 5-23.

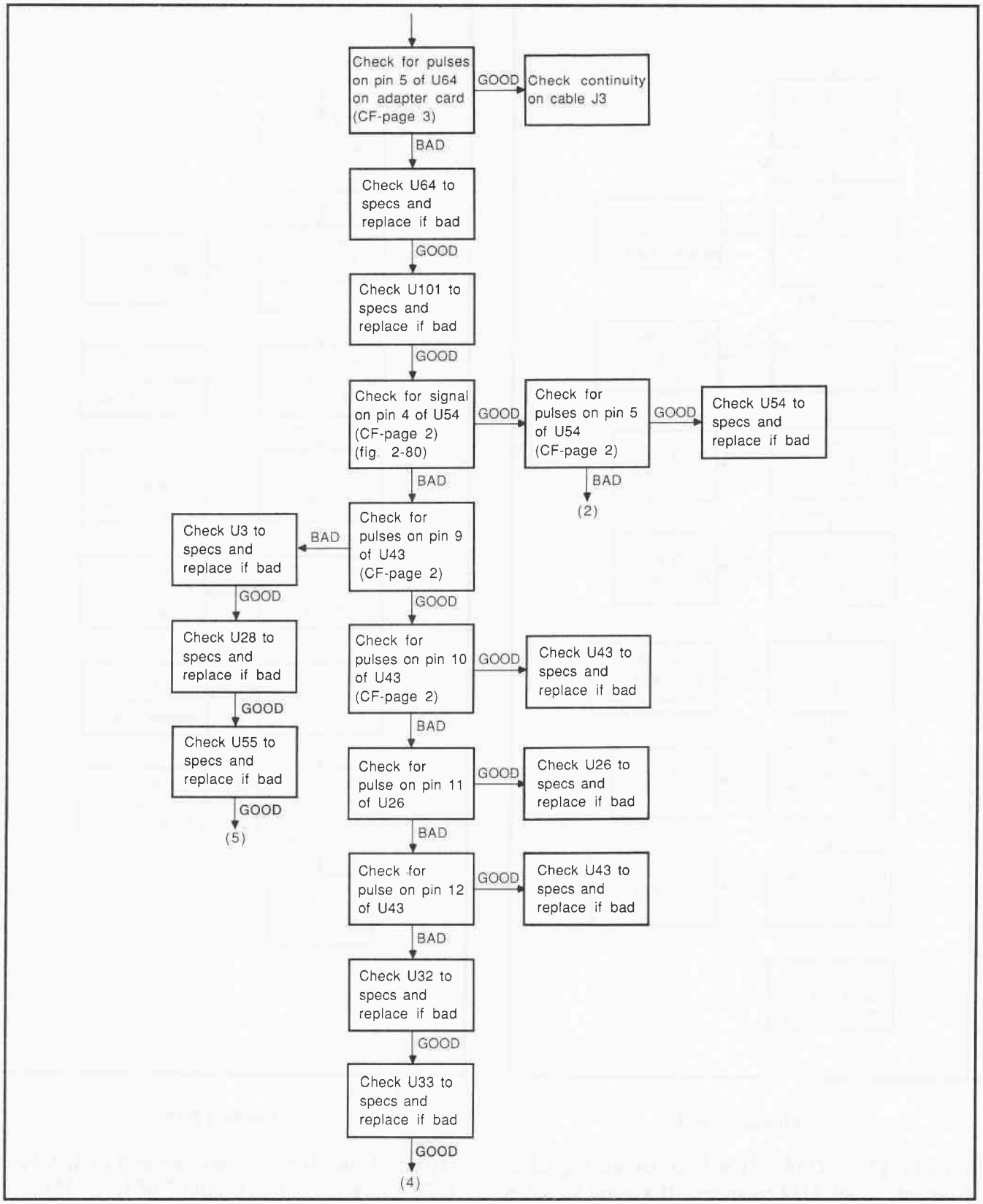


Flowchart 5-24.

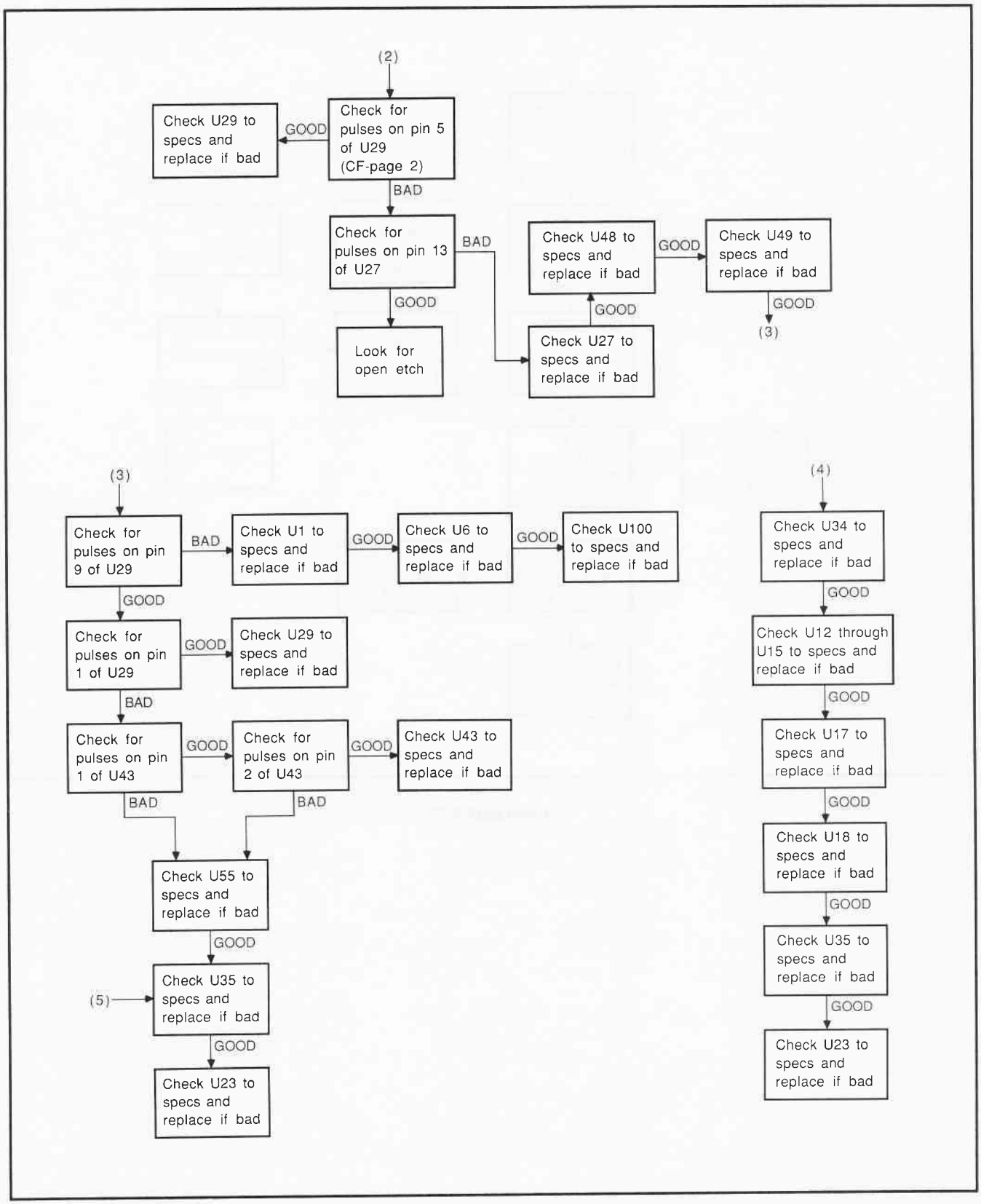


Flowchart 5-25.

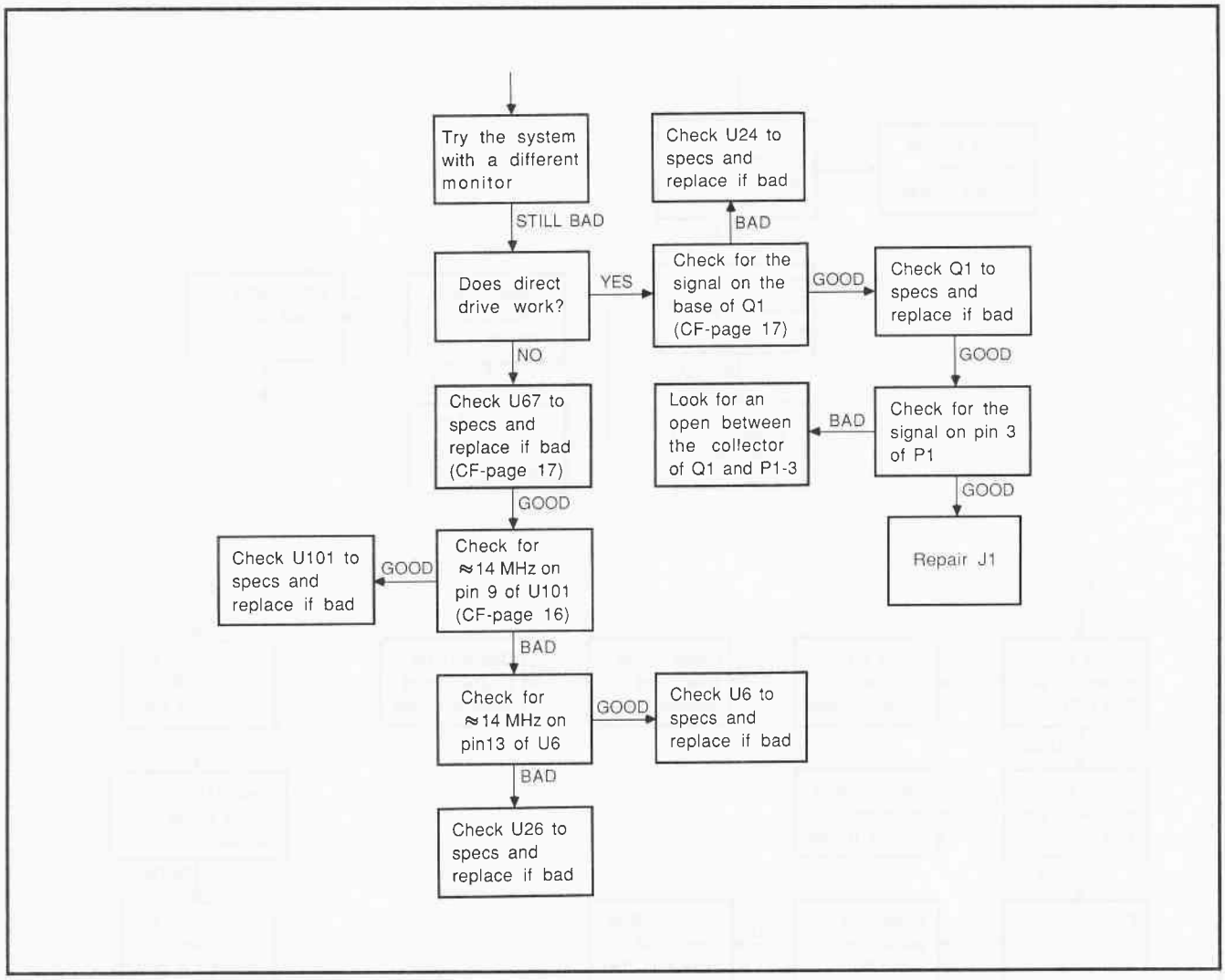
on pin 17 of U34. If a bad, or no signal is present, check U34 to specs. If a good signal is present on pin 17 of U34, check for pulses on pin 12 of U63. If no pulses are found on pin 12, replace U36. If pulses are present on pin 12 of U63, check for pulses on pin 7 of U95. If pulses are present, check U95 to specs. If no pulses are found, check U63 to specs. (See flowchart 5-43.)



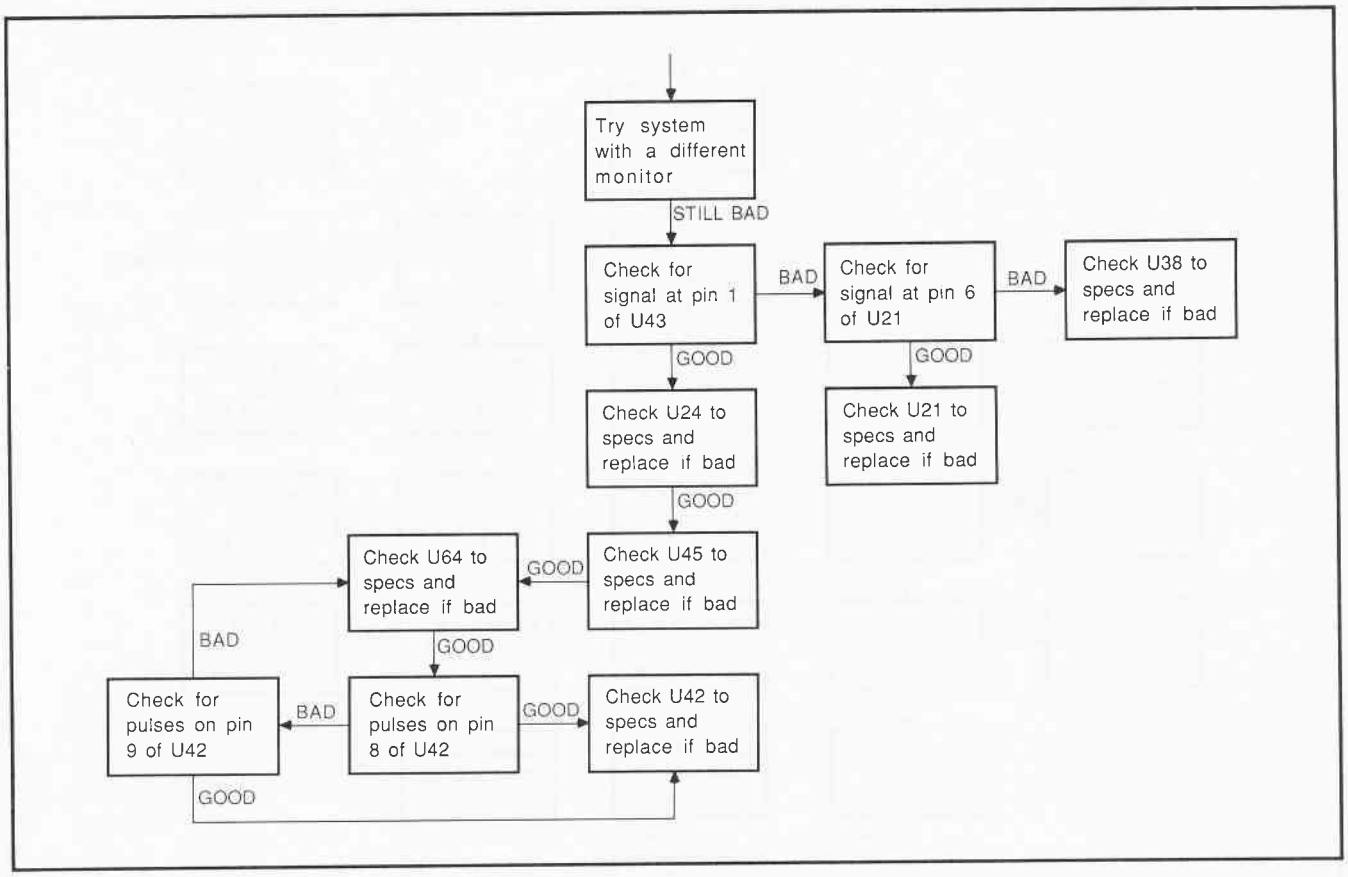
Flowchart 5-26.



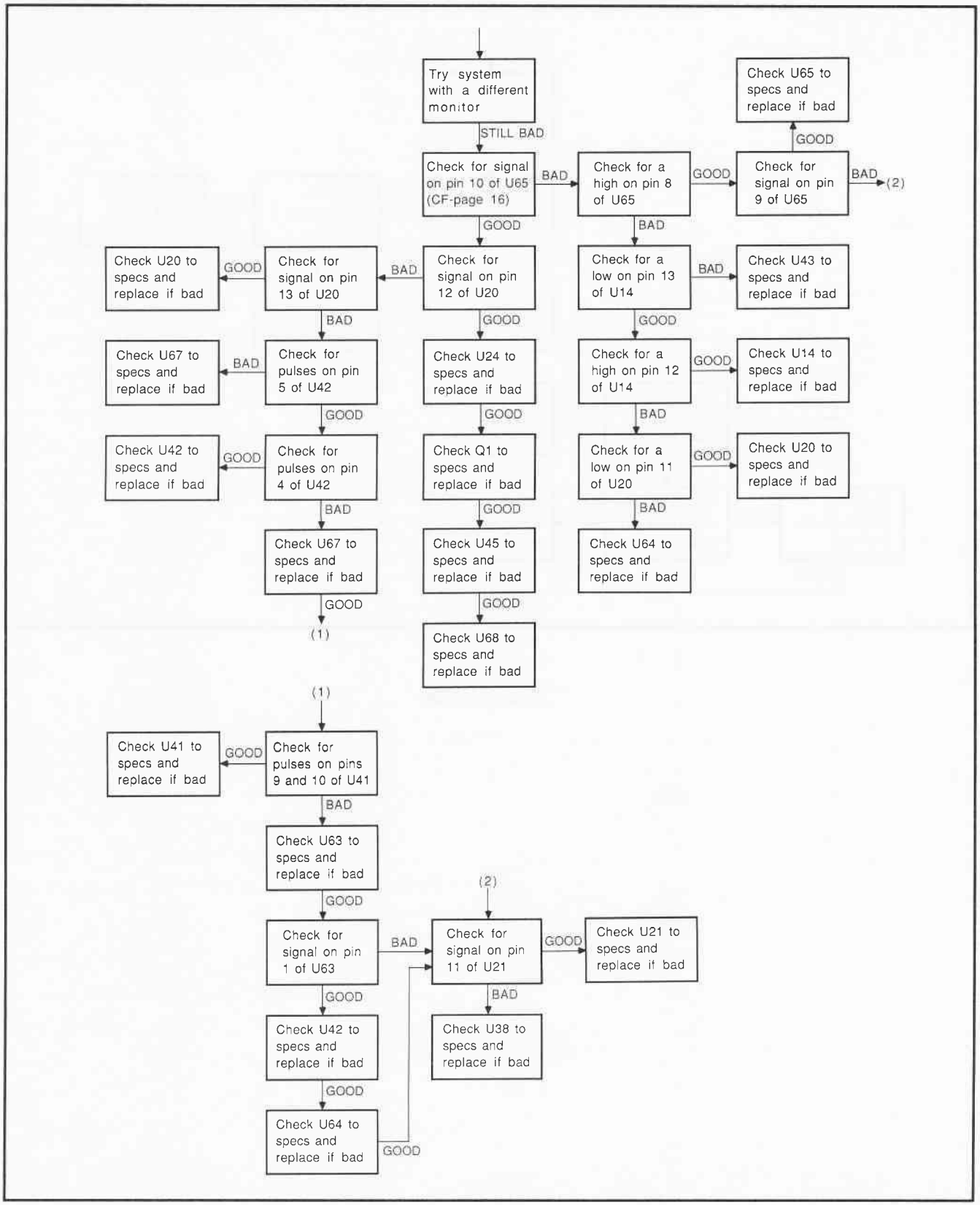
Flowchart 5-26. "cont."



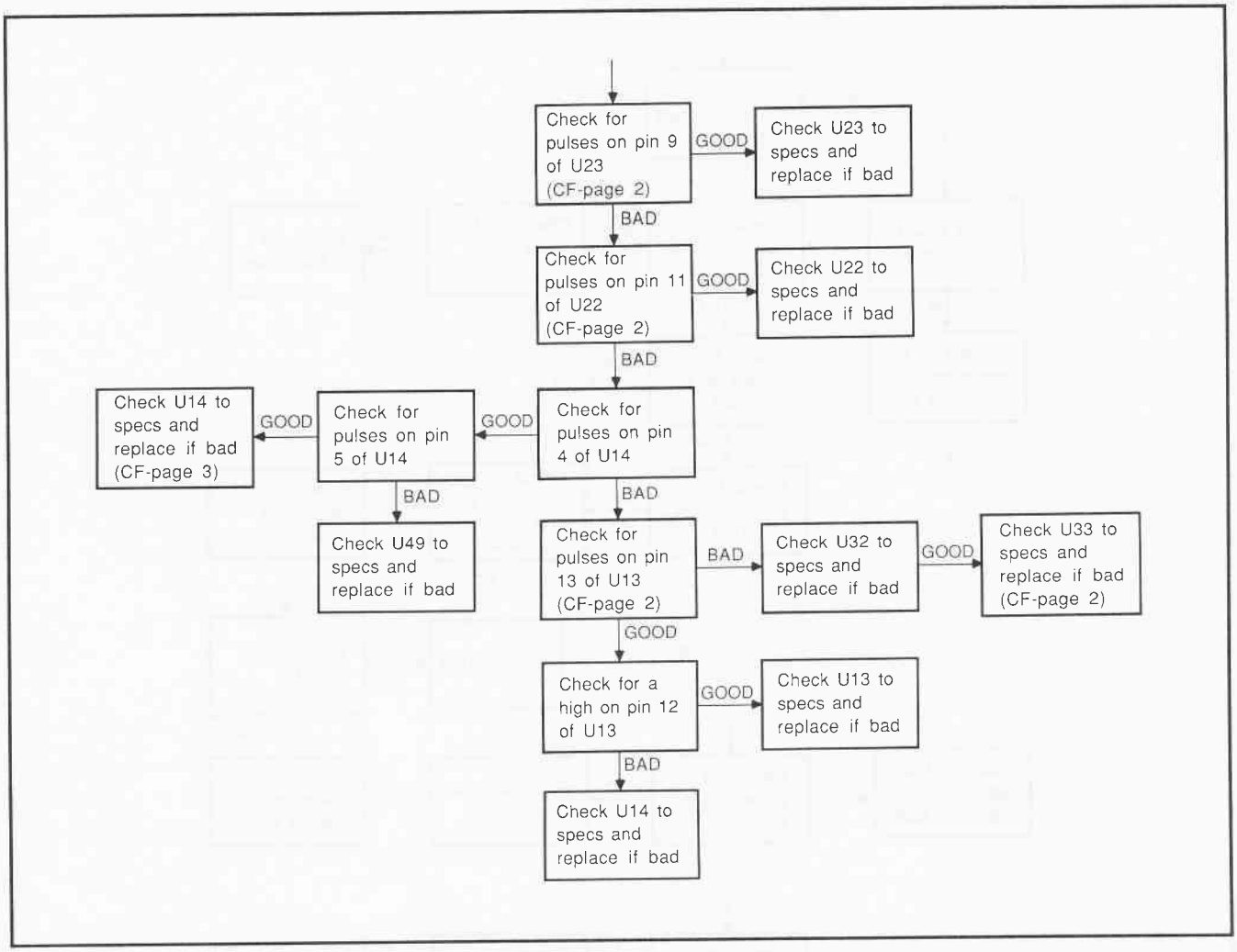
Flowchart 5-27.



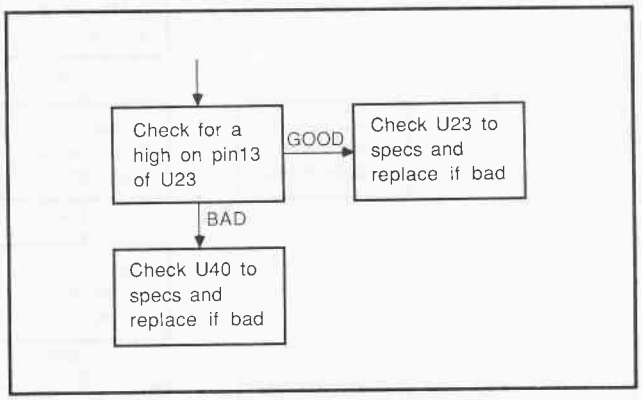
Flowchart 5-28.



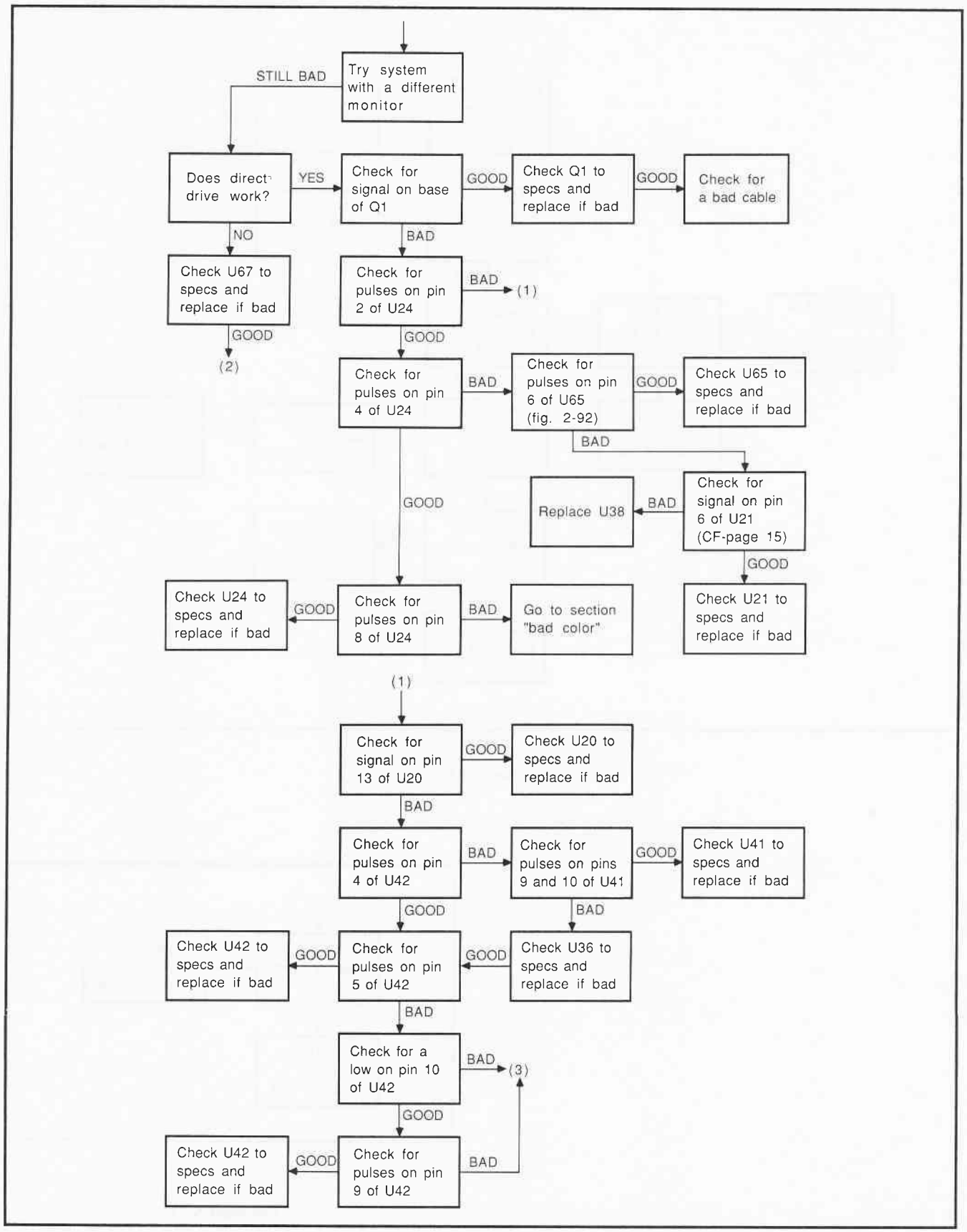
Flowchart 5-29.



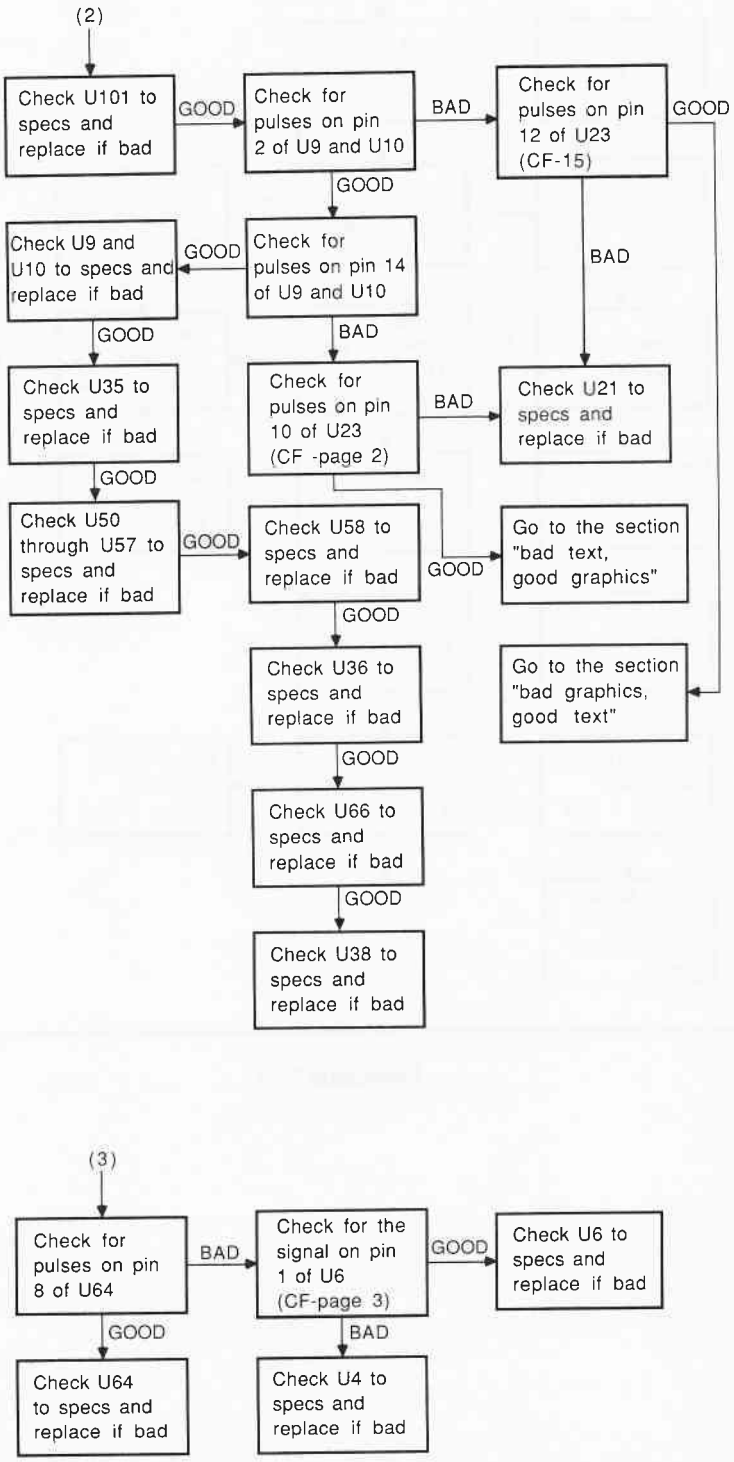
Flowchart 5-30.



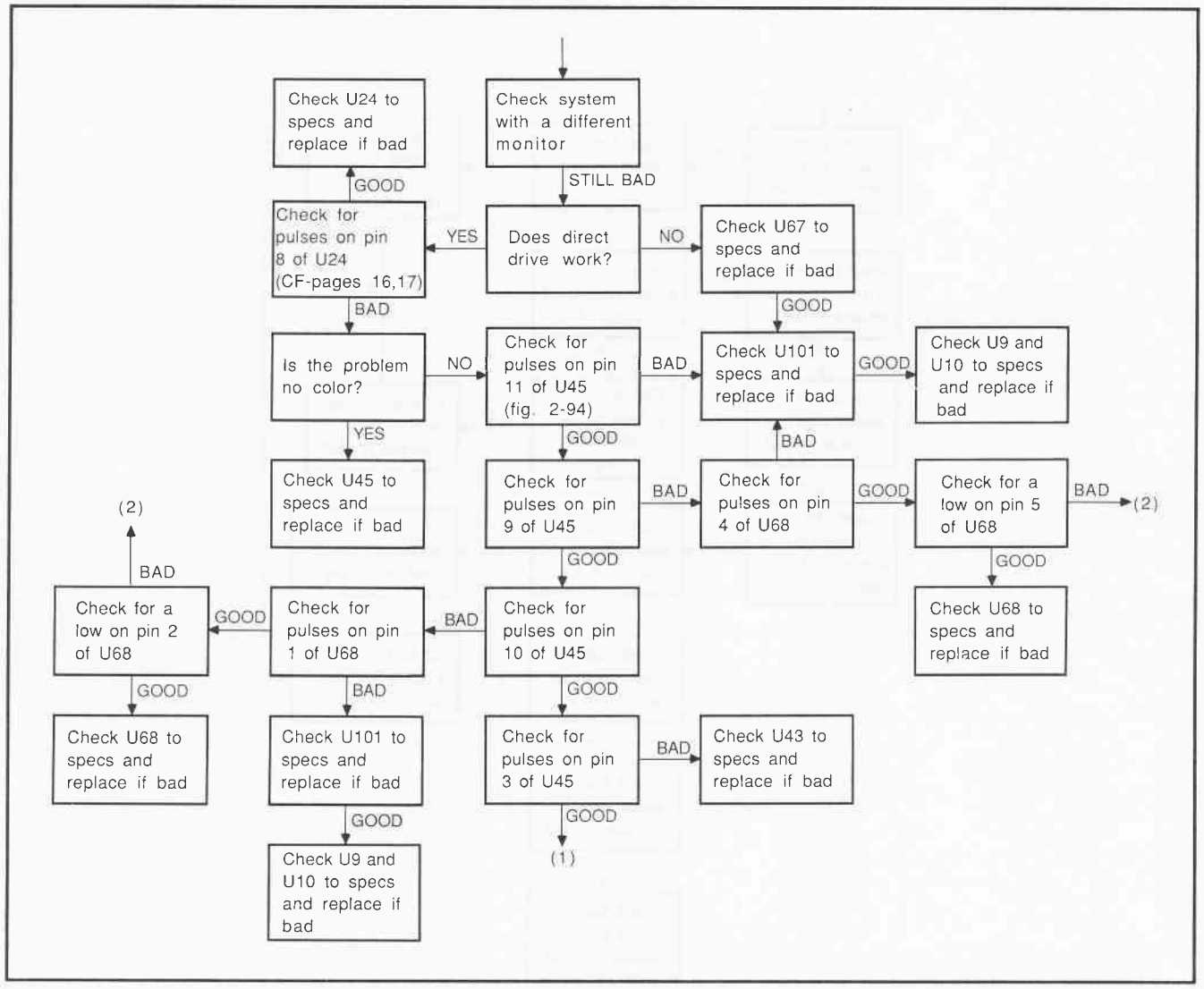
Flowchart 5-31.



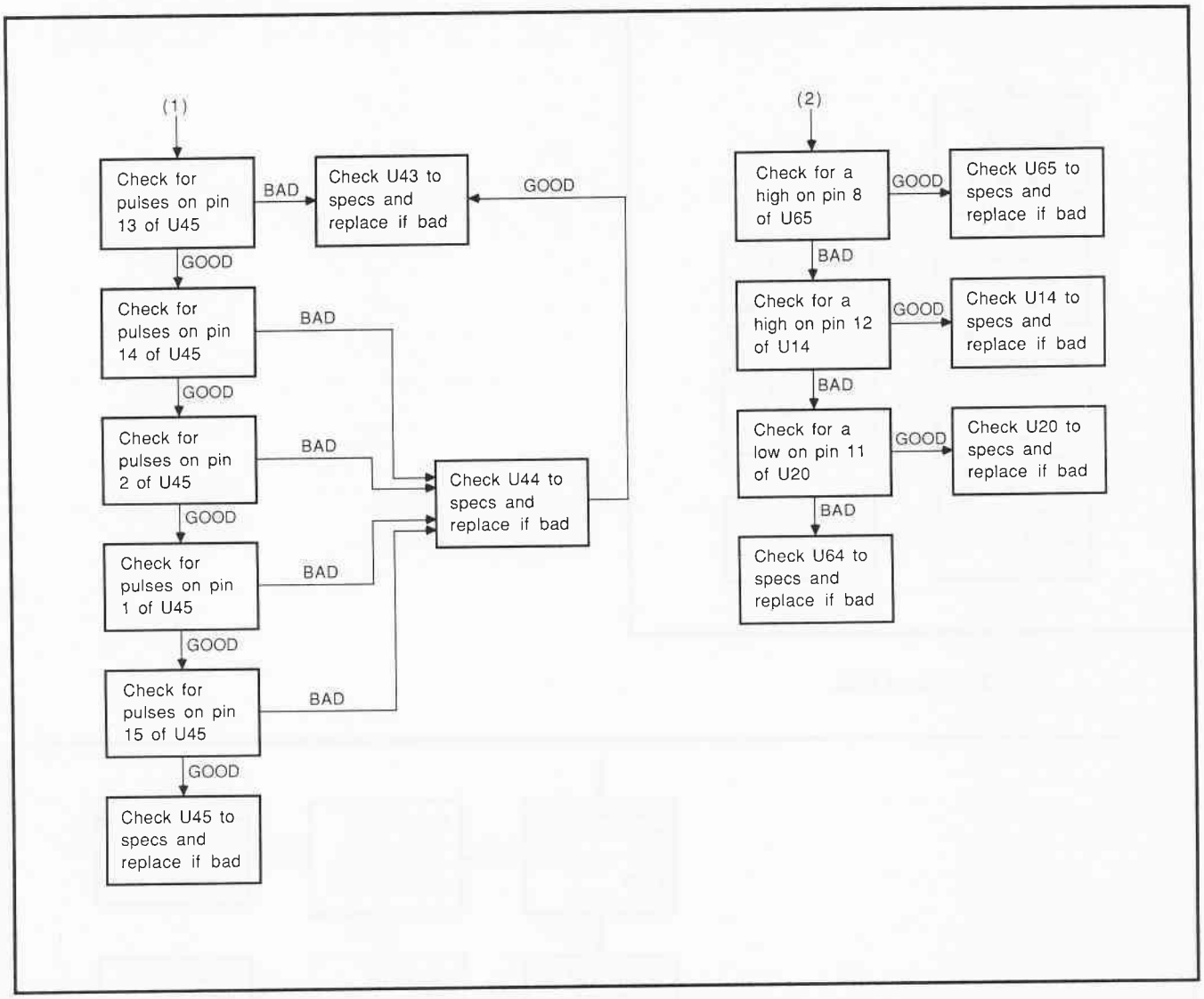
Flowchart 5-32.



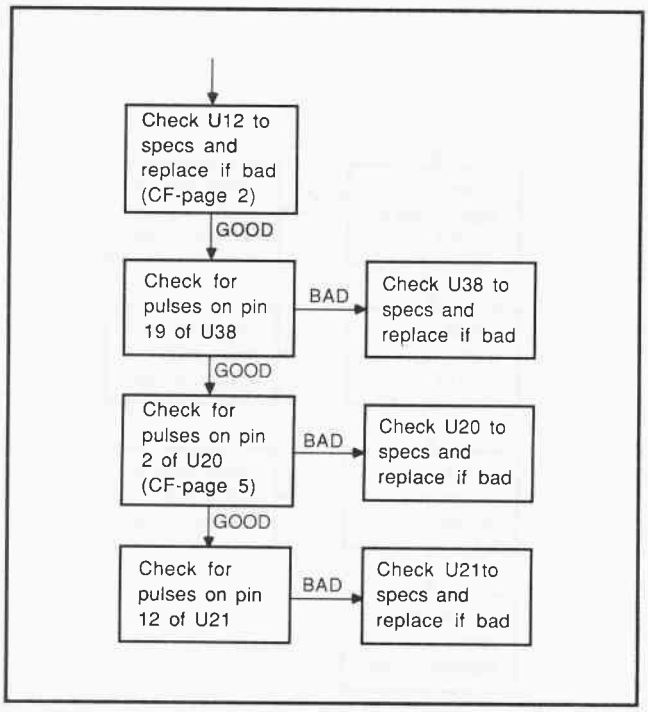
Flowchart 5-32. "cont."



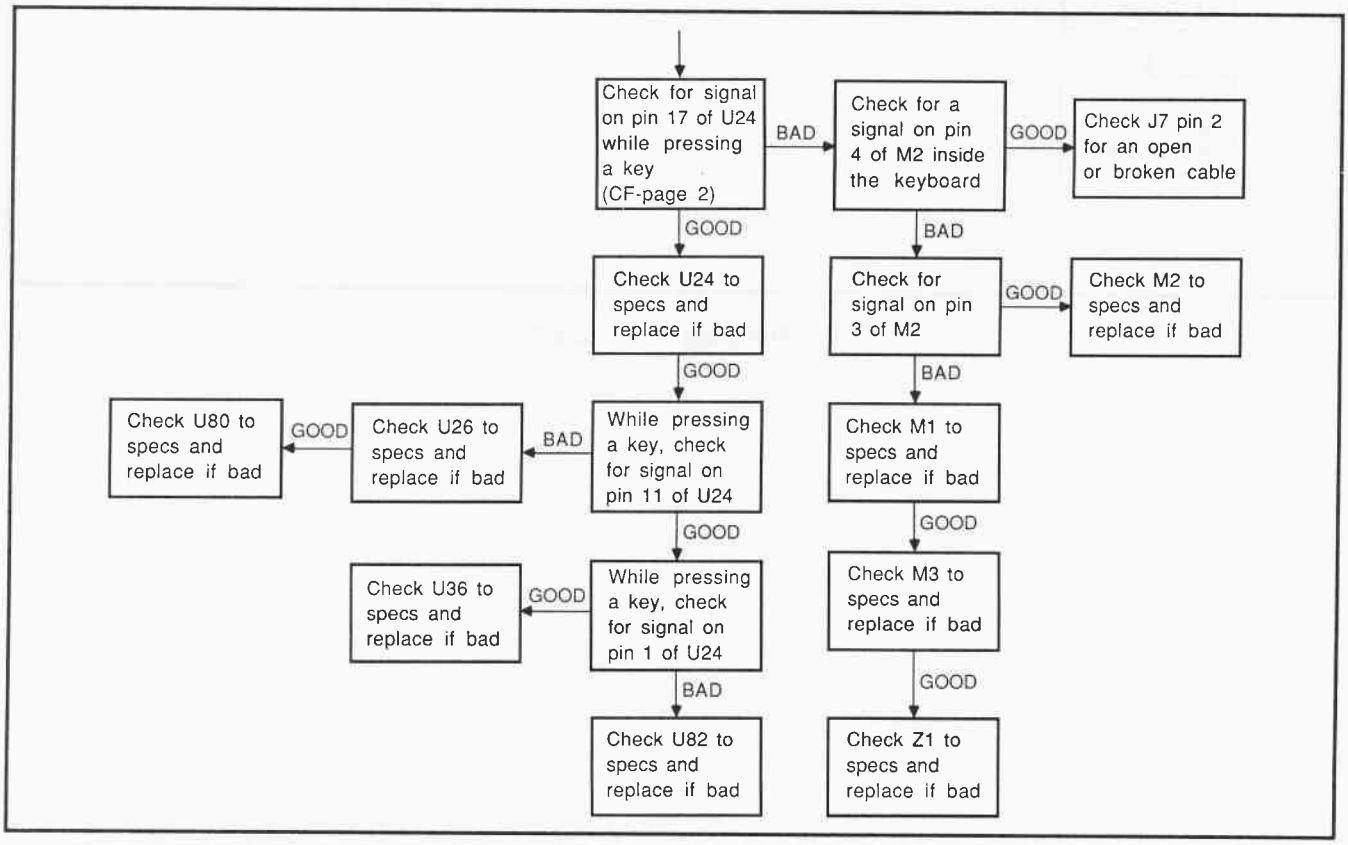
Flowchart 5-33.



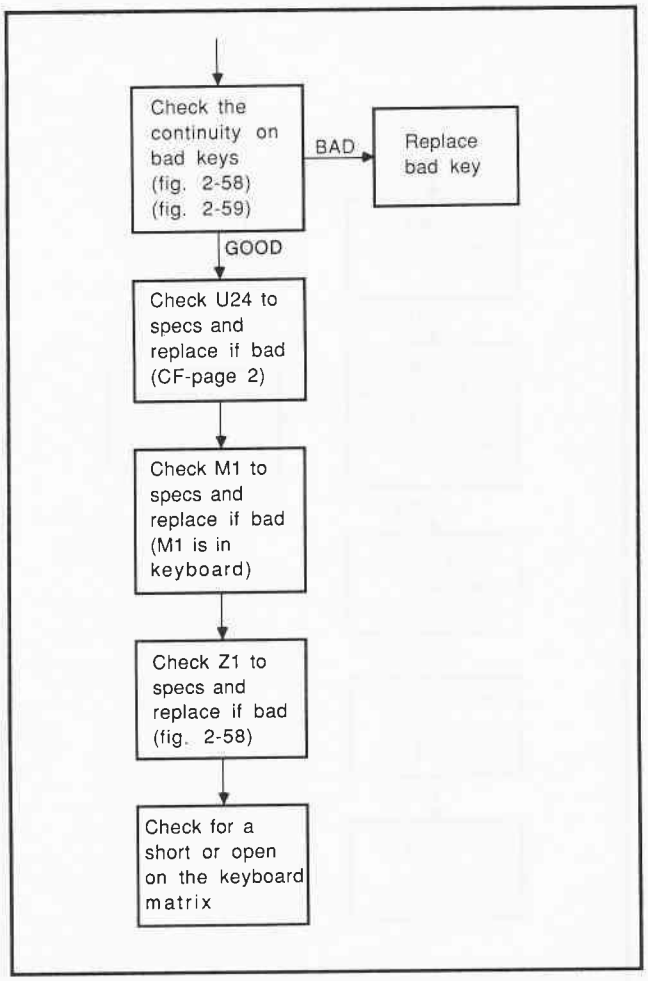
Flowchart 5-33. "cont."



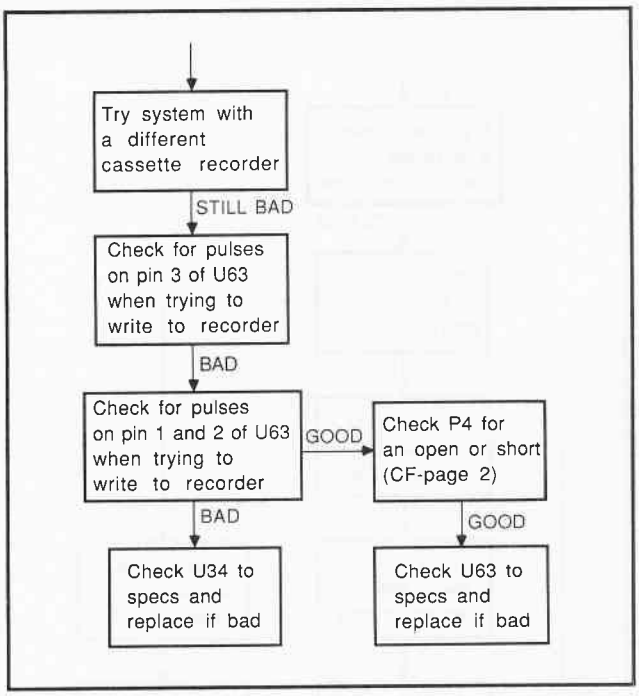
Flowchart 5-34.



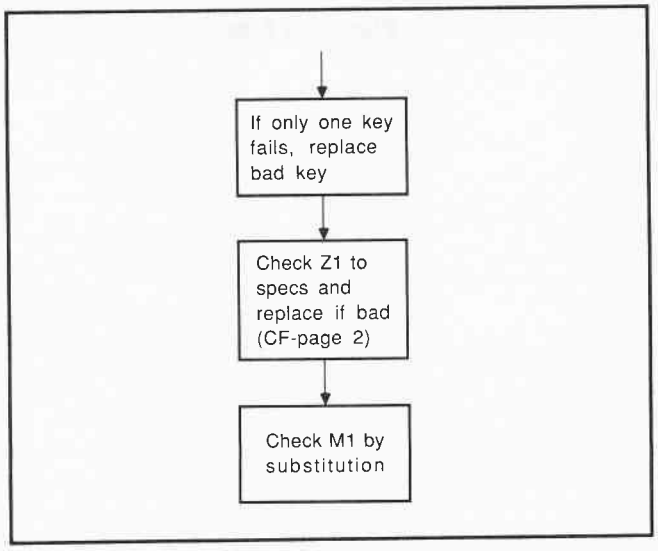
Flowchart 5-35.



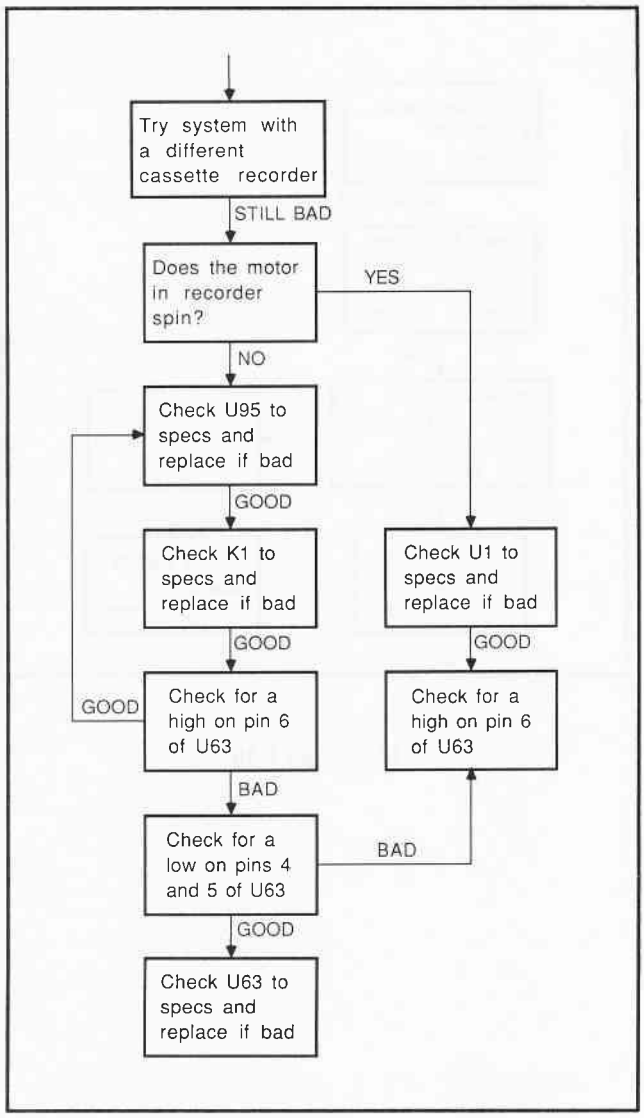
Flowchart 5-36.



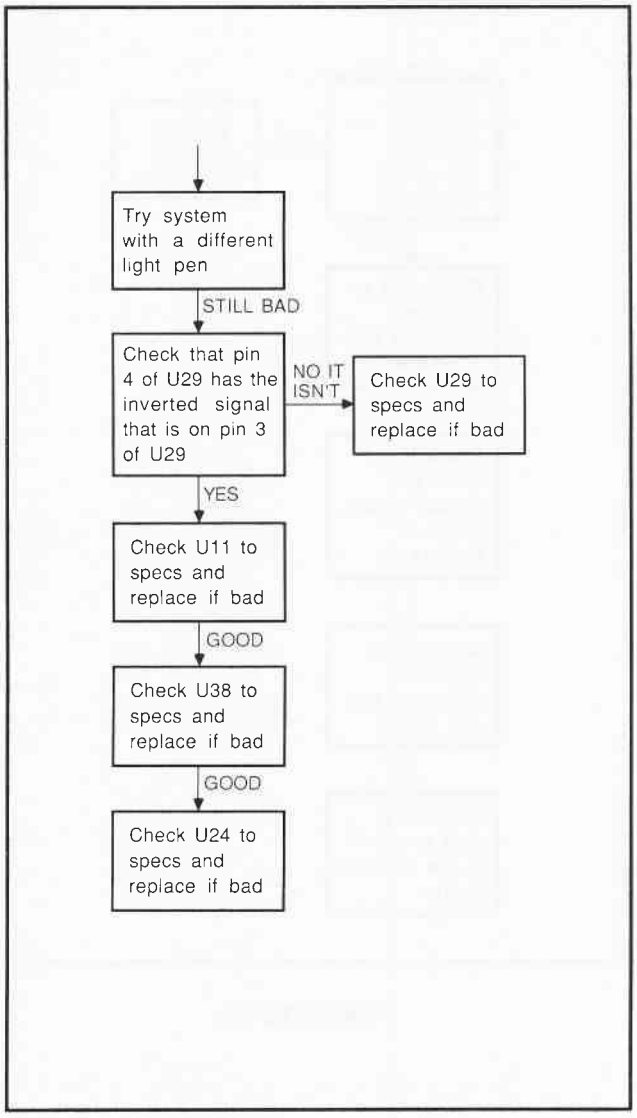
Flowchart 5-38.



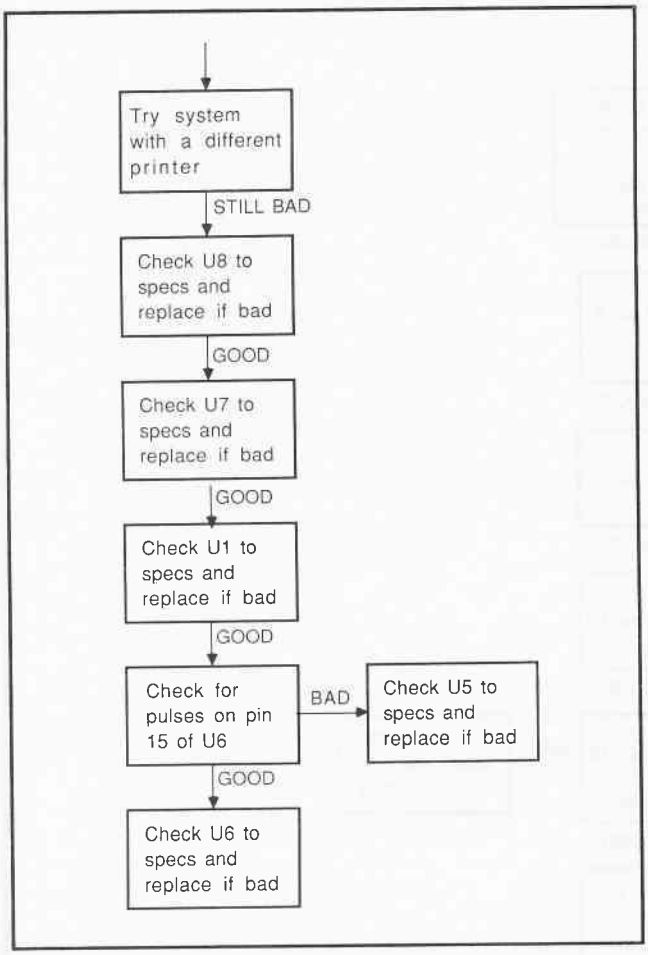
Flowchart 5-37.



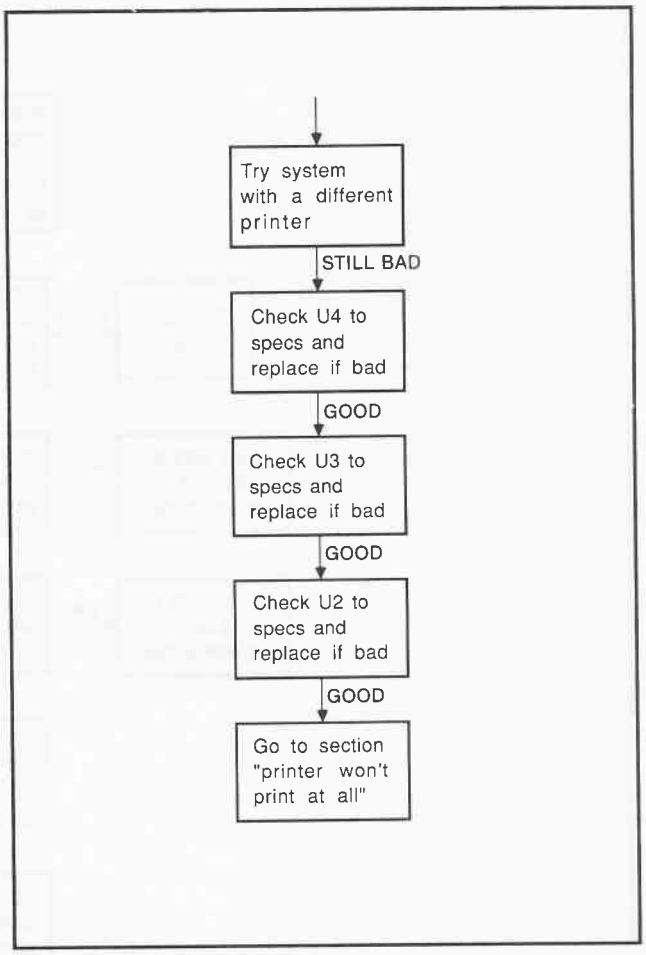
Flowchart 5-39.



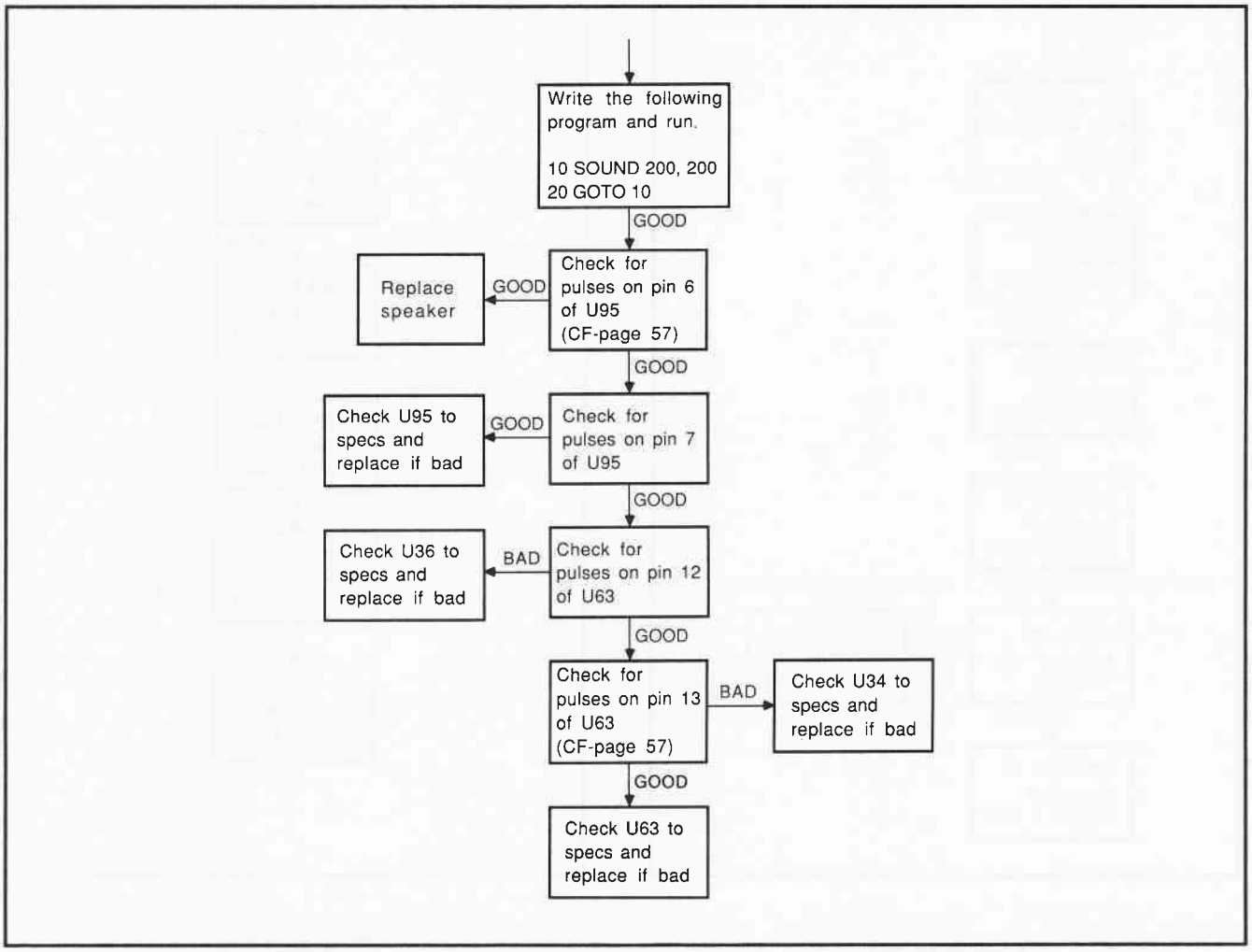
Flowchart 5-40.



Flowchart 5-41.



Flowchart 5-42.



Flowchart 5-43.

A

Data Sheet

Computer: IBM Personal Computer

Manufacturer: IBM Corporation Armonk, New York

Size: 19.6" x 16.1" x 5.5"

Weight: 20.9 lb. without disk drive installed

Power Required: 63.5 watts (maximum)
110/220 volts

CPU: 8088 Microprocessor

Data Word Size: 8 bits

CPU Clock Speed: 4.77 MHz

Memory Size: 40K ROM
16K-bytes on system board
1 Mbyte directly addressable memory

Mass Storage Capability: Two disk drives internally
Two disk drives externally
160K bytes–single density
320K bytes–double density

Keyboard Size: 83 keys
256 character codes

Display: 25 lines of 40 characters, 16 colors
25 lines of 80 characters, 16 colors

Graphics Capability: 100 rows of 160 picture elements (pixels), 16 colors
200 rows of 320 pixels, 4 colors
200 rows of 640 pixels, black and white
32 special graphics characters

Input/Output: Five 62-pin expansion slots
Auxiliary power connection
Cassette connector
Detachable keyboard
2 1/4 inch speaker

Standard Software: Cassette BASIC

Optional Software: PC-DOS
MS-DOS
Cassette BASIC
Advanced BASIC

B

Chip Listing

IBM PC SYSTEM BOARD CHIP LISTING

Label	Integrated Circuit	Description
U1	MC1741	General purpose operational amplifier
U2	8259	Programmable interrupt controller
U3	8088	Microprocessor
U4	8087	Numeric data processor
U5	74LS30	8-input NAND gate
U6	8288	Bus controller
U7	74LS373	Octal transparent latch
U8	74LS245	Tristate octal transceiver
U9	74LS373	Octal transparent latch
U10	74LS373	Octal transparent latch
U11	8284	Clock generator
U12	74LS245	Tristate octal transceiver
U13	74LS245	Tristate octal transceiver
U14	74LS245	Tristate octal transceiver
U15	74LS244	Tristate octal buffer
U16	74LS244	Tristate octal buffer
U17	74LS244	Tristate octal buffer
U18	74LS373	Octal transpartent latch
U19	74LS670	Tristate 4 x 4 register file
U20	RN1	4.7K ohm DIP resistor network
U21	SW1	DIP switch
U22	RN2	2K ohm DIP resistor network
U23	74LS244	Tristate octal buffer
U24	74LS322	8-bit serial/parallel-in register with sign extend
U25	SW2	DIP switch
U26	74LS175	Quad D flip-flop
U27	74LS02	Quad 2-input NOR gate
U28	empty	Spare ROM socket
U29	9264 ROM	8K x 8-bit static ROM
U30	9264 ROM	8K x 8-bit static ROM
U31	9264 ROM	8K x 8-bit static ROM
U32	9264 ROM	8K x 8-bit static ROM
U33	9264 ROM	8K x 8-bit static ROM
U34	8253	Programmable interval timer
U35	8237	DMA controller
U36	8255	Programmable peripheral interface
U37	4164 RAM	64K x 1-bit dynamic RAM
U38	4164 RAM	64K x 1-bit dynamic RAM
U39	4164 RAM	64K x 1-bit dynamic RAM
U40	4164 RAM	64K x 1-bit dynamic RAM
U41	4164 RAM	64K x 1-bit dynamic RAM
U42	4164 RAM	64K x 1-bit dynamic RAM
U43	4164 RAM	64K x 1-bit dynamic RAM
U44	4164 RAM	64K x 1-bit dynamic RAM
U45	4164 RAM	64K x 1-bit dynamic RAM
U46	74LS138	1/8 decoder/demultiplexer
U47	74LS138	1/8 decoder/demultiplexer
U48	74LS138	1/8 decoder/demultiplexer
U49	74LS08	Quad 2-input AND gate
U50	74LS02	Quad 2-input NOR gate
U51	74LS04	Hex inverter
U52	74LS00	Quad 2-input NAND gate
U53	4164 RAM	64K x 1-bit dynamic RAM
U54	4164 RAM	64K x 1-bit dynamic RAM

Label	Integrated Circuit	Description
U55	4164 RAM	64K x 1-bit dynamic RAM
U56	4164 RAM	64K x 1-bit dynamic RAM
U57	4164 RAM	64K x 1-bit dynamic RAM
U58	4164 RAM	64K x 1-bit dynamic RAM
U59	4164 RAM	64K x 1-bit dynamic RAM
U60	4164 RAM	64K x 1-bit dynamic RAM
U61	4164 RAM	64K x 1-bit dynamic RAM
U62	74LS158	Quad 2-input data selector/ multiplexer
U63	74LS38	Quad 2-input NAND buffer
U64	74LS20	Dual 4-input NAND gate
U65	74LS138	1/8 decoder/demultiplexer
U66	74LS138	1/8 decoder/demultiplexer
U67	74LS74	Dual D flip-flop
U68	RN3	4.7K ohm DIP resistor network
U69	4164 RAM	64K x 1-bit dynamic RAM
U70	4164 RAM	64K x 1-bit dynamic RAM
U71	4164 RAM	64K x 1-bit dynamic RAM
U72	4164 RAM	64K x 1-bit dynamic RAM
U73	4164 RAM	64K x 1-bit dynamic RAM
U74	4164 RAM	64K x 1-bit dynamic RAM
U75	4164 RAM	64K x 1-bit dynamic RAM
U76	4164 RAM	64K x 1-bit dynamic RAM
U77	4164 RAM	64K x 1-bit dynamic RAM
U78	RN4	30 ohm DIP resistor network
U79	74LS158	Quad 2-input data selector/ multiplexer
U80	74LS125	Quad tristate buffer
U81	74S00	Quad 2-input NAND gate
U82	74S74	Dual D flip-flop
U83	74LS04	Hex inverter
U84	74LS10	Triple 3-input NAND gate
U85	4164 RAM	64K x 1-bit dynamic RAM
U86	4164 RAM	64K x 1-bit dynamic RAM
U87	4164 RAM	64K x 1-bit dynamic RAM
U88	4164 RAM	64K x 1-bit dynamic RAM
U89	4164 RAM	64K x 1-bit dynamic RAM
U90	4164 RAM	64K x 1-bit dynamic RAM
U91	4164 RAM	64K x 1-bit dynamic RAM
U92	4164 RAM	64K x 1-bit dynamic RAM
U93	4164 RAM	64K x 1-bit dynamic RAM
U94	74LS04	Hex inverter
U95	75477	Relay driver
U96	74LS74	Dual D flip-flop
U97	74S08	Quad 2-input AND gate
U98	74LS175	Quad D flip-flop
U99	74LS04	Hex inverter

Other Components

Location	Device	Description
D1	Type FC	Silicon diode
X1	Crystal	14.31818 MHz crystal oscillator

MONOCHROME MONITOR/ PRINTER ADAPTER CHIP LISTING

Label	Integrated Circuit	Description
U1	74LS74	Dual D edge-triggered flip-flop
U2	74LS04N	Hex inverter
U3	74LS08	Quad 2-input AND gate
U4	74LS74	Dual D edge-triggered flip-flop
U6	74LS10	Triple 3-input NAND gate
U7	74LS139	1-of-8 decoder/demultiplexer
U8-U15	2114	1K x 4-bit RAM
U16-U18	74LS157	Quad 2-to-1 multiplexer
U19	74LS244	3-state octal buffer
U20,21	74LS374	3-state octal D flip-flop
U22	74LS244	3-state octal buffer
U23	74LS245	3-state octal bus transceiver
U24	74LS153	Dual 4-to-1 multiplexer
U25	74LS00	Quad 2-input NAND gate
U26	74S11	Triple 3-input AND gate
U27	74LS02	Quad 2-input NOR gate
U28	74LS393	Dual binary ripple counter
U29	74LS175	Quad D type flip-flop with reset
U30	74LS273	Octal D-type flip-flop with reset
U31	74LS273	Octal D-type flip-flop with reset
U32	74LS166	8-bit parallel-in-serial-out shift register
U33	MK36906	8k character generator ROM
U34	74LS273	Octal D-type flip-flop with reset
U35	MC6845	CRT Controller
U36	74LS125	Quad 3-state buffer
U37	74LS240	Octal 3-state inverter buffer
U38	7405	Open collector hex inverter
U39	74LS174	Hex D flip-flop with reset
U40	74LS244	Octal buffer (3-state)
U41	74LS347	Octal D flip-flop (3-state)
U42	74LS139	Dual 1-of-4 decoder/demulti-plexer
U43	74S32	Quad 2-input OR gate
U44	74LS04	Hex inverter
U45	74LS74	Dual D edge-triggered flip-flop
U46	74LS08	Quad 2-input AND gate
U47	74LS138	1-of-8 decoder/demultiplexer
U48	74LS138	1-of-8 decoder/demultiplexer
U49	74LS138	1-of-8 decoder/demultiplexer
U50	74LS138	1-of-8 decoder/ demultiplexer
U51	74LS138	1-of-8 decoder/demultiplexer
U52	74LS138	1-of-8 decoder/demultiplexer
U53	74LS00	Quad 2-input NAND gate
U54	74S86	Quad 2-input exclusive OR gate
U55	74LS174	Hex D flip-flop with reset
U56	74LS04	Hex inverter
U57	74LS02	Quad 2-input NOR gate
U58	74LS175	Quad D-type flip-flop with reset
U59	74LS125	Quad 3-state buffer
U60	74LS244	Octal buffer (3-state)
U61	74LS155	Dual 1-of-4 decoder/ demultiplexer

Label	Integrated Circuit	Description
U62	74S20	Dual 4-input NAND gate
U63	74LS157	Quad 2-to-1 multiplexer
U64	74LS244	3-state octal buffer
U100	74LS32	Quad 2-input OR gate
U101	74LS74	Dual D edge-triggercd flip-flop

COLOR GRAPHICS ADAPTER CHIP LISTING

Label	Circuit	Description
U1	74S112	Dual J-K negative edge-triggered flip-flop
U2	74LS74	Dual D-type edge-triggered flip-flop
U3	74S86	Quad 2-input exclusive OR gate
U4,U5	74S174	Hex D-type flip-flop with reset
U6	74LS04	Hex inverter
U7,U8	74LS166	8-bit parallel-in-serial-out shift register
U9,U10	74153	Dual 4-to-1 multiplexer
U11	74LS74	Dual D-type edge-triggered flip-flop
U12	74LS393	Dual binary ripple counter
U13	74LS08	Quad 2-input AND gate
U14	74LS32	Quad 2-input OR gate
U15	74LS00	Quad 2-input NAND gate
U16	74LS04	Hex inverter
U17,U18, U19	74LS138	1-of-8 decoder/demultiplexer
U20	74LS04	Hex inverter
U21	74LS174	Hex D flip-flop with reset
U22	74LS51	Dual AND-OR-invert gate
U23	74LS32	Quad 2-input OR gate
U24	74LS244	Octal buffer (3-state)
U25	74LS00	Quad 2-input NAND gate
U26	74S04	Hex inverter
U27	74LS51	Dual 2-wide 2-input AND-OR -inverter gate
U28	74LS10	Triple 3-input NAND gate
U29	74S04	Hex inverter
U30	74LS32	Quad 2-input OR gate
U31	74S08	Quad 2-input AND gate
U32	74LS166	8-bit parallel-in-serial-out shift register
U33	8340 (MK36000)	8k character generator ROM
U34,U35	74LS273	Octal D-type flip-flop with reset
U36	74LS244	Octal buffer (3-state)
U37	74LS374	Octal D-type flip-flop (3-state)
U38	46505 (6845)	CRT controller
U39,U40	74LS174	Hex D-type flip-flop with reset
U41	74LS08	Quad 2-input AND gate
U42	74LS86	Quad 2-input exclusive OR gate
U43	74S74	Dual D edge-triggered flip-flop
U44	74S74	Dual D edge-triggered flip-flop
U45	74LS151	8-to-1 multiplexer
U46	74LS00	Quad 2-input NAND gate
U47,U48, U49	74LS51	Dual 2-wide 2-input AND-OR inverter gate
U50-U57	MK4516N-12 (2118-4)	16K X 1-bit RAM
U58-U61	74LS374	3-state octal D flip-flop
U63	74LS175	Quad D flip-flop with reset
U64	74LS164	8-bit serial-in-parallel-out shift register
U65	74LS02	Quad 2-input NOR gate
U66	74LS245	3-state octal bus transceiver
U67	74LS244	3-state octal buffer
U68	74LS86	Quad 2-input exclusive OR gate
U101	74S174	Hex D flip-flop with reset

C

Line Definitions

Label	Location	Definition
A0-A9	Disk drive	Address line bits 0 through 9
A0-A11	Monochrome	Address line bits 0 through 11
A0-A13	Color adapter	Address line bits 0 through 13
A15-A19	Color adapter Monochrome	Address line bits 15 through 19
A0-A19	System board	Address line bits 0 through 19
ACK	System board Monochrome	Acknowledge
AD0-AD7	System board	Buffered address/data bits 0 through 7
ADSTB	System board	Address strobe
ADDR SEL	System board	Address select
AEN	System board Color adapter Disk drive Monochrome	Address enable
AEN BRD	System board	Address enable board
ALE	System board	Address latch enable
ALPHA DOTS	Monochrome	Alpha dots
AT LATCH	Color adapter	Attribute latch
AT0-AT7	Color adapter Monochrome	Attribute bits 0 through 7
AUTO FD XT	System board Monochrome	Auto feed external lines
B(0),B(7)	Monochrome	Bits 0 and 7
BA0	Color adapter	Buffered address bit 0
BA0-BA3	Monochrome	Buffered address bits 0 through 3
BA8-BA19	System board	Buffered address lines 8 through 19
BACK-GROUND 1	Color adapter	Color background one
BD0-BD7	System board Monochrome	Buffered data lines 0 through 7
BLINK	Monochrome	Blink
BLUE	Color adapter	Color blue
BMEMR	Color adapter	Buffered memory read
BUSY	System board Monochrome	Busy
BW,BW1,BW2	Color adapter	Buffered write lines
B/W VIDEO	Monochrome	Black/white video
CA0-CA11	Color adapter	Column address lines
CACS CCLK	Monochrome	Control address chip select control clock
CAS	System board Color adapter	Column address strobe
CAS0-CAS3	System board	Column address strobe lines 0 through 3
CAS CC	Color adapter	Column address strobe closed circuit
CASS DATA IN	System board	Cassette data in
CC LATC	Color adapter	Closed circuit latch
CC0-CC7	Color adapter Monochrome	Closed circuit character bits 0 through 7 Character code lines
CCLK	Color adapter Monochrome	Control clock
CEROM	Monochrome	Chip enable ROM
CGB0	Monochrome	
CLK	System board Color adapter Disk drive Monochrome	Clock

Label	Location	Definition
CLK88	System board	Clock 8088
CLR	Disk drive	Clear
CLR S/R	Color adapter	Clear shift register
CLRVIDEO	Monochrome	Clear video
COLOR SEL	Color adapter	Color select
CPI	Disk drive	Clock pulse
CPU MEM SEL	Color adapter	Central processing unit memory select
CPUMSEL	Monochrome	CPU memory select
CS	Color adapter	Chip select
CS2-CS7	System board	Chip select lines 2 through 7
CURSOR	Color adapter Monochrome	Cursor select
CURSOR BLINK	Monochrome	Cursor blink
CURSOR DLY	Color adapter Monochrome	Cursor delay
CYAN	Color adapter	Color cyan
D0-D7	System board Color adapter Disk drive Monochrome	Data lines 0 through 7
DACK	Disk drive	DMA acknowledge
DACK0-DACK3	System board	DMA acknowledge lines 0 through 3
DACK0 BRD	System board	DMA acknowledge 0 board
DACK 2	Disk drive	DMA acknowledge 2
DACK & TC	Disk drive	DMA acknowledge & terminal count
DATA0-DATA7	Monochrome	Data lines 0-7
DATA GATE	Color adapter	Data gate
DATA IN	Monochrome	Data in
DATA OUT	System board	Data out
DCLK	System board	Data clock
DIR	System board	Direction
DIR (A&B)	System board	Direction (disk drive) A & B
DIR (C&D)	Disk drive	Direction (disk drive) C & D
DISPEN	Disk drive Color adapter	Display enable
DISPEN DLY	Monochrome Color adapter	Display enable delay
DMA	Monochrome Disk drive	Direct memory access
DMA AEN	System board	Direct memory access address enable
DMA CS	System board	Direct memory access chip select
DMA WAIT	System board	Direct memory access wait
DOT CLK	Color adapter Monochrome	Dot clock
DRIVE SELECT (A&B)(C&D)	Disk drive	Drive select (disk drive) A,B,C,D
DRQ0-DRQ3	System board	DMA request 0 through 3
DRQ2	Disk drive	Data request 2
DT/R	System board	Data transmit/receive
E	Color adapter Monochrome	
ENABLE BLINK	Color adapter Monochrome	Enable blink
ENABLE DRIVE (C & D)	Disk drive	Enable drive (disk drive) C, D
EN CPU CAS ADDR	Color adapter	Enable central processing unit column address strobe adder
EN CPU RAS ADDR	Color adapter	Enable central processing unit row address strobe adder
EN CRT CAS ADDR	Color adapter	Enable cathode ray tube column address strobe adder
EN CRT RAS ADDR	Color adapter	Enable cathode ray tube row address strobe adder
EN I/O CK	System board	Enable input/output check
EN I/O CLK	System board	Enable input/output clock
ENB RAM PCK	System board	Enable RAM peripheral check
ERASE	Disk drive	Erase
ERROR	System board Monochrome	Error
F(0),F(1)	Monochrome	
G	System board	Enable
GRPDCD	Monochrome	
GRPH	Color adapter	Graphics
GRPH EN	Color adapter	Graphics enable
H CLK	Color adapter	Horizontal clock
HIGH LIGHT	Monochrome	High light
HIGH RES	Color adapter	High resolution
HM	Disk drive	
HOLDA	System board	Hold access
HORIZ DRIVE	Monochrome	Horizontal drive
HORIZ SYNC	Color adapter	Horizontal synchronization
HORIZ SYNC DLY	Color adapter	Horizontal sync delay
HRES	Monochrome	High resolution
HSYNC	Monochrome	Horizontal synchronization
HSYNC DLY	Monochrome	Horizontal sync delay
I(B),I(F)	Monochrome	Intensity (blink), (full)
INDEX	Disk drive	Index (mark)
INDEX (A&B)	Disk drive	Index (disk drive) A and B
INDEX (C&D)	Disk drive	Index (disk drive) C and D
INIT	System board Disk drive Monochrome	Initialize
INT	System board	Interrupt
INTA	System board	Initialize address
INTR CS	System board	Internal chip select
INT WRT BUSY	Disk drive	Interrupt write busy
I/O CH CK	System board	Input/output channel check
I/O CH CLK	System board	Input/output channel clock
I/O CH RDY	Color adapter Monochrome	Input/output channel ready
IOR	System board Color adapter Disk drive Monochrome	Input/output read
IOW	System board Color adapter Disk drive Monochrome	Input/output write

Label	Location	Definition
IRQ	Disk drive	Interrupt request
IRQ7	Monochrome	Interrupt request 7
IRQ0-IRQ7	System board	Interrupt request lines 0 through 7
IRQEN	Monochrome	Interrupt request enable
IRQ EN	System board	Interrupt request enable
JUMPER	Monochrome	Jumper
LCC5-LCC7	Monochrome	
L CLK	Color adapter	Light clock
L PEN INPUT	Color adapter	Light pen input
L PEN STR	Color adapter	Light pen strobe
L PEN SW	Color adapter	Light pen switch
LOCK	System board	Lock
LVIDEO	Monochrome	Line video
MA0-MA6	Color adapter	Memory address lines 0 through 6
MA0-MA7	System board	Memory address lines 0 through 7
MA0-MA10	Monochrome	Memory address lines 0 through 10
MD0-MD7	System board Color adapter	Memory data lines 0 through 7
MD00-MD11	System board System board	Matrix data lines 00 through 11
MDP		Memory data parity
ME	System board	Memory enable
MEMR	System board Color adapter Monochrome	Memory read
MEMW	System board Color adapter Monochrome	Memory write
MOTOR CNTRL	System board	Motor control
MOTOR ENABLE (A&B)(C&D)	Disk drive	Motor enable (disk drive) A,B,C,D
MOTOR OFF	System board	Motor off
MOTOR ON	Disk drive	Motor on
MR	Disk drive	Memory read
MRQ DMA	System board	Memory request direct memory access
MUX A, MUX B	Color adapter	Multiplexer A, B
NMI	System board	Non-maskable interrupt
NODSPLY	Monochrome	No display
NP INSTL SW	System board	Numeric processor installation switch
NPNPI	System board	Numeric processor numeric processor interrupt
OSC	System board Color adapter	Oscillator
OUT	Disk drive	Output
OVERSCAN B	Color adapter	Overscan blue
OVERSCAN G	Color adapter	Overscan green
OVERSCAN R	Color adapter	Overscan red
OVERSCAN L	Color adapter	Overscan luminance
PB6,PB7	System board	8255 port B bits 6 and 7
PCK	System board	Peripheral check
PCLK	System board	Peripheral clock

Label	Location	Definition
PE	System board Monochrome	Paper end
PPICS	System board	
PWR GOOD	System board	Power good
Q1,Q2,Q3,Q4	Color adapter	Accumulator extension lines 1 through 4
Q5	Monochrome	Accumulator extension line 5
QS0,QS1	System board	Q... status bits 0 and 1
RA0-RA2	Color adapter Monochrome	Read address lines 0 through 2
RA0-RA3	System board	Row address lines 0 through 3
RAM ADDR SEL	System board	RAM address select
RAS	Color adapter	Row address strobe
RAS0-RAS3	System board	Row address strobe bits 0 through 3
RD GATE	Color adapter	Read gate
RDGATEAT	Monochrome	Read gate attribute
RDGATECC	Monochrome	Read gate character code
RDY/WAIT	System board	Ready/wait
RDY TO DMA	System board	Ready to direct memory access
READ DATA	Disk drive	Read data
READ DATA (A&B)(C&D)	Disk drive	Read data (disk drive) A,B,C,D
READY	System board	Ready
RED	Color adapter	Red
REFRESH GATE	System board	Refresh gate
REQIN	System board	Request in
REQOUT	System board	Request out
RESET	System board Disk drive Monochrome	Reset
RESET C	Color adapter	Reset control
RESET DRV	System board Color adapter Monochrome	Reset drive
RMA0-RMA9	Monochrome	Read memory address lines
ROMA 11	Monochrome	ROM address line 11
ROM ADDR SEL	System board	ROM address select
RPA	System board	Read printer data
RPB	System board	Read printer control
RPC	System board	Read printer status
RPA-RPC	Monochrome	Read printer data, control, status
RQ/GT	System board	Request/grant
RVV	Monochrome	Reverse video
S0-S2	System board	Status bits 0 through 2
S0	Disk drive	Side zero
S1	Disk drive	Side 1
SA CLOSED	System board	
S DOTS	Monochrome	Serial dots
SEEK	Disk drive	Seek
SELECT0-SELECT2	System board	Select lines 0 through 2

Label	Location	Definition
SELECT HEAD (A&B)(C&D)	Disk drive	Select head (disk drive) A,B,C,D
SEL BLUE	Color adapter	Select color blue
SEL 1	Monochrome	Select line 1
SEL1, SEL2	Color adapter	Select lines 1 and 2
SENSE A-SENSE H	System board	Sense lines A through H
SERDATA	Monochrome	Serial data
SERIAL DATA	System board	Serial data
SERIN	Monochrome	Serial in
S/L	Color adapter Monochrome	Serial/line
SLCT	System board Monochrome	Select
SLCTIN	Monochrome	Select input
SLCT IN	System board	Select input
SP/EN	System board	Slave program/enable buffer
SPKR DATA	System board	Speaker data
STATUS SEL	Color adapter Monochrome	Status select
STEP (A&B)	Disk drive	Step (disk drive) A,B
STEP (C&D)	Disk drive	Step (disk drive) C,D
STR	Color adapter	Strobe
STROBE	System board Monochrome	Strobe
T/C	System board	Terminal count
TC	System board Disk drive	Terminal count
TC CS	System board	TC chip select
TD0-TD7	Color adapter	Transceiver data lines 0 through 7
TIM2 GATE SPK	System board	Timer 2 gate speaker
TIMER/ CNTR2	System board	Timer control 2
TRACK 0	Disk drive	Trace 0
TRACK 0 (A&B)	Disk drive	Track 0 (disk drive) A and B
TRACK 0 (C&D)	Disk drive	Track 0 (disk drive) C and D
UNDERLINE	Monochrome	Underline
VCO SYNC	Disk drive	Voltage controlled oscillator sync
VERT DRIVE	Monochrome	Vertical drive
VERT SYNC	Color adapter	Vertical synchronization
VERT SYNC DLY	Color adapter	Vertical sync delay
VIDEO	Monochrome	Video
VIDEO ENABLE	Color adapter	Video enable
VSYNC	Monochrome	Vertical synchronization
VSYNC DLY	Monochrome	Vertical sync delay
WE	System board Color adapter Monochrome	Write enable
WPA	System board Monochrome	Write printer data
WPC	System board Monochrome	Write printer control
WR DATA (A&B)	Disk drive	Write data (disk drive) A and B
WR DATA (C&D)	Disk drive	Write data (disk drive) C and D
WRITE	Disk drive	Write
WRITE DATA 00	Disk drive	Write data 00
WRITE DATA 01	Disk drive	Write data 01
WRITE GATE (A&B)(C&D)	Disk drive	Write gate (disk drive) A,B,C,D
WRITE PROTECT	Disk drive	Write protect
WRITE PROTECT (A&B)(C&D)	Disk drive	Write protect (disk drive) A,B,C,D
WRT DMA PG REG	System board	Write direct memory access page register
WRT NMI REG	System board	Write non-maskable interrupt register
WRT TRAN	Disk drive	
XA0-XA12	System board	Buffered address lines
XACK	Color adapter Monochrome	Buffered acknowledge
XD0-XD7	System board	Buffered data lines
XIOR	System board Monochrome	Buffered I/O read
XIOW	System board Monochrome	Buffered I/O write
XMEMR	System board Monochrome	Buffered memory read
XMEMW	System board	Buffered memory write
YELLOW BURST	Color adapter	Yellow burst
2 MHz	Disk drive	2 MHz clock frequency
3.58 MHz	Color adapter	3.58 MHz color oscillator frequency
7 MHz	Color adapter	7 MHz oscillator frequency
14 MHz	Color adapter	14 MHz oscillator frequency
16 MHz	Disk drive	16 MHz system clock

D

Disassembly Procedures

SYSTEM UNIT DISASSEMBLY INSTRUCTIONS

These procedures apply to those repairs that require access to the internal subassemblies of the IBM PC system unit.

System Board Access

Tools and Equipment Required

No. 2 flathead screwdriver

Uncluttered workspace

Container to hold screws until reassembly

Procedure for System Board Access

1. Turn the power off.
2. Unplug the power cord and any peripherals from the rear of the computer.
3. Position the system unit so the rear is facing you.
4. Using a flathead screwdriver remove the 5 screws from the rear plate (see Fig. D-1). (The older model IBM PCs have only two screws on the back plate.)

Fig. D-1. Remove these screws to disassemble the IBM PC.

5. Position the system unit so that the front is facing you.
6. Place your hands on either side of the cover and slide the cover off of the main unit pulling towards you as shown in Fig. D-2.

Procedure for Removing System Board

1. Follow steps 1 through 6 in procedure for system board access.

2. Remove all peripheral cards from the system board.

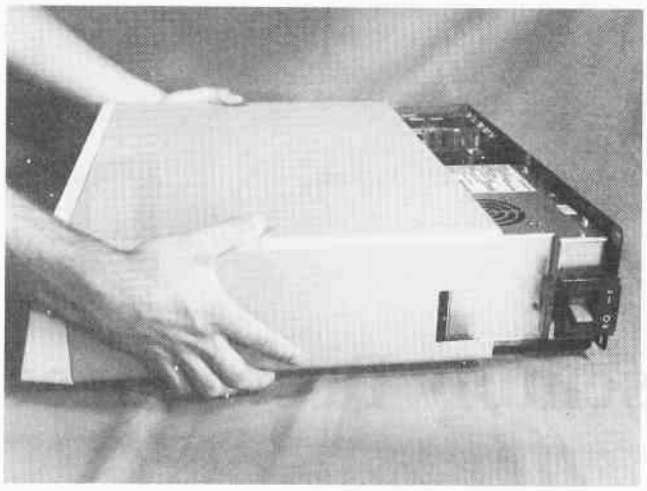


Fig. D-2. Gently slide the system unit cover forward.

3. Remove the power connector from the system board. (This is located in the back right side looking from the front.)
4. Remove the speaker cable from the connector on the lower middle section on the system board.
5. Remove the system board mounting screws as shown in Fig. D-3.
6. Slide the system unit board away from the power supply approximately 2 inches until standoffs can be lifted from their mounting slots.
7. Lift the system board from the system unit.

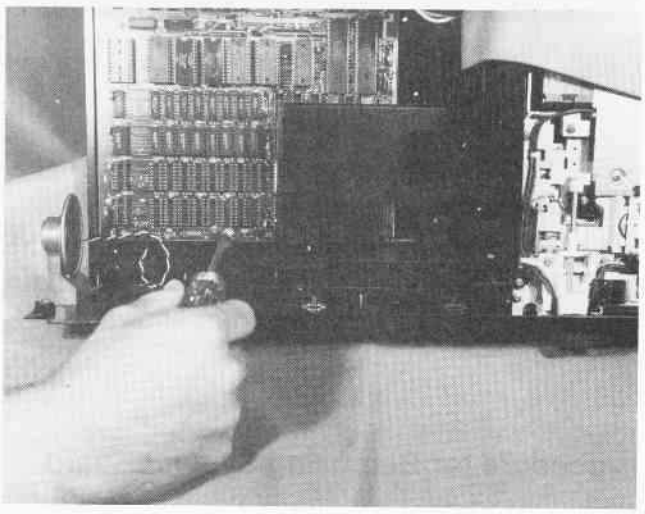


Fig. D-3. Remove the system board mounting screws.

KEYBOARD DISASSEMBLY

This section covers the proper procedures for disassembling the keyboard.

Tools and Equipment Required

Small Phillips-head screwdriver
Uncluttered workspace
Container to hold screws until reassembly

1. Turn the system unit off.
2. Remove the keyboard from the connector in the back of the system unit.
3. Turn the keyboard upside down.
4. Remove the two Phillips-head screws from the bottom of the keyboard plate.
5. Lift the top of the plate up and out of the retaining slots in the chassis of the keyboard.
6. Disconnect the cable from the keyboard assembly.
7. Lift the rear of the keyboard out of the chassis.

POWER SUPPLY REMOVAL

This section describes the steps required to remove the power supply from the chassis.

Tools and Equipment Required

No. 2 flathead screwdriver
Uncluttered workspace
Container to hold screws until reassembly

1. Turn the power off.

2. Unplug the power cord and any peripherals from the rear of the computer.
3. Position the system unit so the rear is facing you.
4. Remove the system unit cover. (See procedure for system board access.)
5. Remove the power connector from the system board. (This is located in the back right side looking from the front.)
6. If you have drives hooked to the system, disconnect the power cables going to the drive analog cards.
7. Remove the four power supply screws on the back of the chassis.
8. Push the power supply forward about $^1/_2$ inch.
9. To remove the supply, lift up and pull the power supply away from the motherboard.

E

Reassembly Procedures

SYSTEM UNIT REASSEMBLY INSTRUCTIONS

Now that the repair is complete, follow these steps to put the system back together.

Tools and Equipment Required

No. 2 flathead screwdriver
Uncluttered workspace
Container to hold screws until reassembly

REINSTALLING SYSTEM BOARD

1. Position all the standoffs hooked to the system board above the mounting holes.
2. Gently push the system board toward the power supply until you can see that the mounting screw holes line up.
3. Reinstall the mounting screws in the system board.
4. Reconnect the signal wires to the speaker.
5. Install the adapter cards.
6. Reconnect the system board power supply connectors.

REASSEMBLING SYSTEM UNIT CASE

1. Gently slide the system unit case forward over the system unit.
2. Reinstall the five flathead screws on the back of the chassis. (The older model IBM PC has only two screws on the back plate.)
3. Reconnect all peripherals and the power cord.

KEYBOARD REASSEMBLY

This section covers the proper procedures for putting the keyboard back together.

Tools and Equipment Required

Small Phillips-head screwdriver
Uncluttered workspace
Container to hold screws until reassembly

1. Position the front of the keyboard assembly into the front of the keyboard chassis.

2. Lower the back of the keyboard down into the chassis.
3. Reconnect the cable to the keyboard assembly.
4. Put the tabs on the front of the base into the slots on the front of the keyboard chassis.
5. Slowly lower the back down–don't forget to include the adjustable legs on the bottom of the keyboard.
6. Install the two Phillips-head screws into the mounting holes on the bottom of the keyboard.
7. Reconnect the cable to the system unit assembly.
8. Power up and test.

POWER SUPPLY INSTALLATION

This section describes the steps needed to reinstall a power supply in the chassis.

Tools and Equipment Required

No. 2 flathead screwdriver

Uncluttered workspace

Container to hold screws until reassembly

1. Hold the power supply unit approximately 1/2 inch from the rear of the chassis, and push the supply toward the motherboard and then back to align the screw holes in the chassis.
2. Replace the four mounting screws for the power supply.
3. Reconnect the disk drive power supply connectors.
4. Reconnect the motherboard power supply connectors.
5. Reconnect the power cord.
6. Power up and test.

> Note: Disk drive disassembly and reassembly instructions are covered in Chapter 3.

F

Replacing Surface Mounted Components

Desoldering and soldering on the IBM PC system board is not easy--the board construction is such that damage to the board traces and solder points can easily occur if you aren't extremely careful.

Caution: Proceed at your own risk.

1. Reread the section on soldering techniques found in Chapter 3.
2. Be sure you're using a temperature-controlled iron.
3. Disassemble the machine and remove the system board.
4. Place the board on its edge and locate the component to be replaced.
5. If possible, during chip removal, attach an extractor tool to the component to be replaced and use a DIP tip on your iron to heat the pins and remove the chip.
6. Or use a vacuum solder "sucker," or braided wick and the temperature controlled iron to heat the pins (start at the corners first, then desolder every other pin to avoid overheating one area of the board trace) until the component comes free.
7. Clean the solder holes using the techniques described in Chapter 3.
8. If a chip was removed, install an IC socket in its place on the system board. This lets you install a replacement chip into an already soldered connection eliminating the need to solder directly to the chip pins themselves.
9. If a transistor is being replaced, install a transistor socket in the system board connection holes.
10. If a resistor, diode, or capacitor is being replaced, solder the leads directly in the opened holes in the board.
11. Reinstall the system board in the computer's housing.
12. Reassemble the computer.
13. Reconnect the power cord.
14. Power up and test.
15. Return the computer to service (or break back down again to replace another possibly faulty component).

Note: If your efforts didn't solve the problem, and you've replaced all the suspected components with good components, you have little recourse—replace the entire system board assembly.

G

ASCII Code Chart

Hexa-decimal	ASCII	Hexa-decimal	ASCII	Hexa-decimal	ASCII	Hexa-decimal	ASCII	Hexa-decimal	ASCII	Hexa-decimal	ASCII
00	^@ (NULL)	15	^U	2B	+	40	@	56	V	6B	k
01	^A	16	^V	2C	,	41	A	57	W	6C	l
02	^B	17	^W	2D	-	42	B	58	X	6D	m
03	^C	18	^X	2E	.	43	C	59	Y	6E	n
04	^D	19	^Y	2F	/	44	D	5A	Z	6F	o
05	^E	1A	^Z	30	0	45	E	5B	[	70	p
06	^F	1B	^[(ESCAPE)	31	1	46	F	5C	\	71	q
07	^G (BELL)	1C	^\	32	2	47	G	5D	]	72	r
08	^H (BACKSPACE)	1D	^]	33	3	48	H	5E	^'	73	s
09	^I (TAB)	1E	^^	34	4	49	I	5F	_	74	t
0A	^J (LINEFEED)	1F	^	35	5	4A	J	60	`	75	u
0B	^K	20	SPACE	36	6	4B	K	61	a	76	v
0C	^L	21	!	37	7	4C	L	62	b	77	w
0D	^M	22	"	38	8	4D	M	63	c	78	x
0E	^N	23	#	39	9	4E	N	64	d	79	y
0F	^O	24	$	3A	:	4F	O	65	e	7A	z
10	^P	25	%	3B	;	50	P	66	f	7B	{
11	^Q	26	&	3C	<	51	Q	67	g	7C	|
12	^R	27	'	3D	=	52	R	68	h	7D	}
13	^S	28	(	3E	>	53	S	69	i	7E	~
14	^T	29	)	3F	?	54	T	6A	j	7F	DELETE
		2A	*			55	U				

(The symbol ^ represents a control character.)

H

Hexadecimal to Decimal Conversion Chart

Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec	Hex	Dec
$00	00	$1D	29	$3A	58	$57	87	$74	116	$91	145	$AE	174	$CD	205	$EA	234
$01	01	$1E	30	$3B	59	$58	88	$75	117	$92	146	$AF	175	$CE	206	$EB	235
$02	02	$1F	31	$3C	60	$59	89	$76	118	$93	147	$B0	176	$CF	207	$EC	236
$03	03	$20	32	$3D	61	$5A	90	$77	119	$94	148	$B1	177	$D0	208	$ED	237
$04	04	$21	33	$3E	62	$5B	91	$78	120	$95	149	$B2	178	$D1	209	$EE	238
$05	05	$22	34	$3F	63	$5C	92	$79	121	$96	150	$B5	181	$D2	210	$EF	239
$06	06	$23	35	$40	64	$5D	93	$7A	122	$97	151	$B6	182	$D3	211	$F0	240
$07	07	$24	36	$41	65	$5E	94	$7B	123	$98	152	$B7	183	$D4	212	$F1	241
$08	08	$25	37	$42	66	$5F	95	$7C	124	$99	153	$B8	184	$D5	213	$F2	242
$09	09	$26	38	$43	67	$60	96	$7D	125	$9A	154	$B9	185	$D6	214	$F3	243
$0A	10	$27	39	$44	68	$61	97	$7E	126	$9B	155	$BA	186	$D7	215	$F4	244
$0B	11	$28	40	$45	69	$62	98	$7F	127	$9C	156	$BB	187	$D8	216	$F5	245
$0C	12	$29	41	$46	70	$63	99	$80	128	$9D	157	$BC	188	$D9	217	$F6	246
$0D	13	$2A	42	$47	71	$64	100	$81	129	$9E	158	$BD	189	$DA	218	$F7	247
$0E	14	$2B	43	$48	72	$65	101	$82	130	$9F	159	$BE	190	$DB	219	$F8	248
$0F	15	$2C	44	$49	73	$66	102	$83	131	$A0	160	$BF	191	$DC	220	$F9	249
$10	16	$2D	45	$4A	74	$67	103	$84	132	$A1	161	$C0	192	$DD	221	$FA	250
$11	17	$2E	46	$4B	75	$68	104	$85	133	$A2	162	$C1	193	$DE	222	$FB	251
$12	18	$2F	47	$4C	76	$69	105	$86	134	$A3	163	$C2	194	$DF	223	$FC	252
$13	19	$30	48	$4D	77	$6A	106	$87	135	$A4	164	$C3	195	$E0	224	$FD	253
$14	20	$31	49	$4E	78	$6B	107	$88	136	$A5	165	$C4	196	$E1	225	$FE	254
$15	21	$32	50	$4F	79	$6C	108	$89	137	$A6	166	$C5	197	$E2	226	$FF	255
$16	22	$33	51	$50	80	$6D	109	$8A	138	$A7	167	$C6	198	$E3	227		
$17	23	$34	52	$51	81	$6E	110	$8B	139	$A8	168	$C7	199	$E4	228		
$18	24	$35	53	$52	82	$6F	111	$8C	140	$A9	169	$C8	200	$E5	229		
$19	25	$36	54	$53	83	$70	112	$8D	141	$AA	170	$C9	201	$E6	230		
$1A	26	$37	55	$54	84	$71	113	$8E	142	$AB	171	$CA	202	$E7	231		
$1B	27	$38	56	$55	85	$72	114	$8F	143	$AC	172	$CB	203	$E8	232		
$1C	28	$39	57	$56	86	$73	115	$90	144	$AD	173	$CC	204	$E9	233		

Routine Preventive Maintenance

Preventive maintenance (PM) is one of the least used techniques for operational cost reduction, yet the savings that result can be substantial. If the equipment doesn't fail, you can't evaluate the bottom-line savings in conducting proper PM. But after the first mind-boggling repair expense, the fact will sink in. This failure might have been prevented by doing some easy, routine maintenance.

Someone once said "Time is money." Failure to take the time to do routine preventive maintenance can indeed cost money. Do your PMs!

Many manufacturers are not sure what optimum PM should be. Others prefer you don't do any PMs. (The effect is to cause more equipment repair jobs for you.) Among those who recommend PMs, they vary in recommended PM schedules for similar hardware (for example, disk drives).

The listing that follows is a consensus of recommendations of manufacturers, dealers, users, and the author's own experience.

OPTIMUM PM SCHEDULE

Modify the schedule if intermittents occur frequently.

Daily

Log Operational Time

Estimate disk drive "run-light-on" time.
Estimate printer "printing" time.
Estimate computer "power-on" time.

Monitor Humidity

This is a measure of static electricity.

Weekly

Clean Computer System Work Area

Pick up all loose trash, reshelve scattered books, restore magazines, toss out old printed paper,

toss those "bad" disks you've been saving, wipe down hardware with an antistatic, dust-absorbing cloth, wipe desk and bench space with antistatic cloth, and vacuum shelves, desk, and floor.

Clean Equipment Housings and Cases

Wipe chassis with antistatic cloth, "wash" with lightly soaped damp cloth.

Clean Display Screens

Use antistatic "dust-off" type spray or damp cloth of antistatic solution.

Clean Drive Read Head

Clean drive read head after 40 hours of "run-light-on" use.

Monthly

Some manufacturers recommend that the drive read head be demagnetized after 40 hours "run-light-on" use.

Clean Inside Computer

Disassemble according to the procedures in Appendix D.

Use soft brush and long narrow vacuum cleaner hose nozzle (it helps to spray the nozzle with antistatic first).

Clean Inside Printer

Use same technique as for cleaning inside computer.

Check Ventilation Filters in Equipment

Replace if cleaning is not practical (filter becomes worn or badly soiled).

Check Connector Contacts

Look for signs of corrosion, pitting, or discoloration.

Clean if necessary. The corrosion-removing wipes that also coat the surface with a lubricating coating to protect it from atmospheric corrosion are strongly recommended.

Every Other Month

Reseat All Socketed Chips on the Motherboard and Peripheral Cards

Disassemble according to the procedures in Appendix D.

Disconnect and Reconnect Cable and Connector Plugs

This removes corrosion buildup.

Apply Antistatic Treatment to Computer Work Area

Clean Inside Printer

Use nonmagnetic, plastic vacuum hose nozzle and soft camel hair brush. Spray or wipe nozzle with antistatic spray or solution first.

Every Six Months

Replace Vent Filters

Only if some of the equipment has filters. None are standard in the IBM PC.

Check Disk Drive Speed

Speed test programs are advertised in computer publications.

Remember the room light, strobe mark test (see Chapter 3).

Check Head Alignment

Do this only if you suspect a disk problem.

Clean Connector Contacts

If you haven't done this during earlier inspection checks, conduct this PM now. Do this PM more often if the computer system is used in a smoggy part of the country or near salty air.

Clean Disk Drive Read Head

If the system is used daily, the drive heads may need cleaning about now, but this depends very much on the kind and quality of floppy disks that are used.

Conduct Printer Routine Inspection

Do this every six months or 500,000 lines of print. Check the tightness of the screws and connectors. Conduct a printer self-test as described in the printer owner's manual.

Annually

Take Routine Maintenance Infrared Photo (optional)

Only do this if you're into this form of PM or troubleshooting.

J

Bibliography

Anderson, Garry J. "Designer's Guide to the CMOS STD Bus," *Electronic Products* (November 17, 1983), pp. 81-87.

Archibald, Dale. "The Making of the Magnetic Media for Micros," *Softalk* (February 1982), pp. 160-164.

Babcoke, Carl. "Practical Information About Testing and Replacing Capacitors," *Electronic Servicing* (July 1970), pp. 28-37.

Babcoke, Carl. "Quick Testing of Transistors," *Electronic Servicing* (November 1970), pp. 26-33.

Babcoke, Carl. "Simple Servicing Tips," *Electronic Servicing & Technology* (July 1983), pp. 44-49.

Baker, Alan, and Mielke, Neal. "Detecting Quality and Reliability Defects in EPROMs," *Electronic Test* (November 1983), pp. 56-62.

Barden, William, Jr. "Getting Your Micro Repaired," *Popular Computing* (May 1983), pp. 54-58.

Bausell, James. "Desoldering Components From High Density PCB's," *Electronics* (February 1984), pp. 97-101.

Bausell, James, "Desoldering Components, Using Continuous Vacuum Solder Extraction," *Electronic Servicing & Technology* (May 1987), pp 13-18.

Beenker, F.P.M. "Systematic and Structured Methods for Digital Board Testing," *VLSI Systems Design* (January 1987), pp 50-58.

Belt, Forest. "1-2-3-4 Servicing Simplifies Industrial Electronic Maintenance," *Electronic Servicing* (September 1979), pp. 21-27.

Bohannon, George. "The ABCs of IBM Graphics," *Softalk* (February 1983), pp. 30-36.

Boyd, Alan. "System Notebook," *Softalk* (January 1983), pp. 60-64.

Boyd, Alan. "System Notebook," *Softalk* (February 1983), pp. 82-85.

Boyd, Alan. "System Notebook," *Softalk* (March 1983), pp. 108-110.

Brenner, Robert C. *IBM PC Troubleshooting & Repair Guide.* Howard W. Sams & Company, Indianapolis, Indiana, 1986.

Bristol, Rod. "Believable Time Measurements With Oscilloscopes," *Electronics Test* (October 1986), pp. 49-47.

Camenker, Brian. "The Making of the IBM PC," *Byte* (November 1983), pp. 254-256.

"Caring f or a Personal Computer," *Electronic Servicing & Technology* (June 1985), pp. 42-47.

"Choosing and Using the Proper Soldering Iron," *Electronic Servicing & Technology* (December 1981), pp. 36-39.

Crosby, Mark L. "Singin' the Disk I/O Blues," *Apple Orchard* (Winter 1981/82), pp. 63-68.

Cunningham, John E. "Troubleshooting Digital Equipment," *Electronic Servicing* (September 1980), pp. 18-21.

Curran, Lawrence J., and Shuford, Richard S. "IBM's Estridge," *Byte* (November 1983), pp. 88-97.

Dale, Alan. "1-2-3-4 Servicing," *Electronic Servicing* (December 1970), pp. 26-30.

Dash, Glen "Understanding EMI Test Methods Eases Product Acceptance," *EDN* (May 1983), pp. 183-192.

Davidson, Homer L. "Ten Dogs in TV Repair," *Electronic Servicing & Technology* (September 1984), pp. 12-23.

Davis, Dwight B. "Diagnostics Improve as Computer Systems Proliferate," *Mini-Micro Systems* (August 1982), pp. 115-123.

Develop Test Technology for VHSIC. RADC-TR-83-148, Rome Air Development Center, Air Force Systems Command, Griffiss AFB, New York, September 1983.

DeVoney, Chris. *IBM's Personal Computer*. Que Corporation, Indianapolis, Indiana, 1983.

DeVore, John A. "To Solder Easily," *Circuits Manufacturing* (June 1984), pp. 62-70.

Dvorak, John C. "Let's Modernize the Microcomputer," *PC Magazine* (October 28, 1986), pp. 77.

Earle, A. Scott. "Taking a Closer Look at the RGB Monitor," *PC Magazine* (April 3, 1984), pp. 145-154.

Engel, George M. "Line Cleaner–A Construction Project," *inCider* (August 1983), pp. 108-110.

Fastie, Will. "The IBM Personal Computer," *Creative Computing* (December 1981), pp. 19-40.

Field, Tim. "The IBM PC and the Intel 8087 Coprocessor," *Byte* (August 1983), pp. 331-374.

Field, Tim. "Enhancing Screen Displays for the IBM PC," *Byte* (November 1983), pp. 99-116.

Final Report: The Identification and Assessment of On-Chip Self-Test and Repair Concepts. Naval Electronics Systems Command, September 1981.

Freedman, David H. "Designing the Right Enclosure," *Mini-Micro Systems* (August 1983), pp. 229-242.

Freitag, Walter D. "Lubricants for Separable Connectors," *IEEE Transactions on Parts, Hybrids, and Packaging* (March 1977), p. 32.

Glasco, David B. "Using IBM's Marvelous Keyboard," *Byte* (May 1983), pp. 402-415.

Glinert-Cole, Susan. "Upgrading a PC to an 'Xtra T,'" *PC Tech Journal* (February 1984), pp. 75-82.

Goldblatt, Robert C. "How Computers Can Test Their Own Memories," *Computer Design* (July 1976), pp. 125-129.

Goodman, Robert. "An Ounce of Prevention," *Electronic Servicing & Technology* (May 1983), pp. 24-39.

Goodman, Robert L. "Techniques for Repairing Intermittents," *Electronic Servicing* (July 1979), pp. 33-39.

Goodstein, Max. "Learning From a Tough Dog TV Repair," *Electronic Servicing & Technology* (April 1987), pp. 25, 57.

Grolle, Carl G. *Electronic Technician's Handbook of Time-Savers and Shortcuts*. Parker Publishing Company, Inc., West Nyack, New York, 1974.

Hancock, Earle. "A Man of Letters," *inCider* (December 1983), pp. 172-174.

Hancock, Earle. "Do-It-Yourself Disk Drive Repair," *inCider* (November 1983), pp. 32-34.

Harwood, Robert. "Diagnostic and Utility Software," *Personal Computing* (October 1981), pp. 47-54, 166-169.

Hogan, Thom. "We're Not in Kansas Anymore," *The Portable Companion* (June/July 1982), pp. 11-14.

Howson, Hugh R. "POKEing Around in the IBM PC," *Byte* (November 1983), pp. 121-131.

Hunter, David. "The Roots of DOS," *Softalk* (March 1983), pp. 12-15.

Illowsky, Dan, and Abrash, Michael. "Up, Down, Right, Left & Check," *PC Tech Journal* (February 1984), pp. 93-116.

Izen, Bud. "Microcomputer Troubleshooting: Components of a Personal Computer," *Electronic Servicing & Technology* (November 1985), pp. 22-27.

Jesson, Joseph E. Smart Keyboards Help Eliminate Entry Errors," Computer Design (October 1982), pp. 137-142.

Kaminer, David A. "What to Do When Your System Crashes," *Popular Computing* (April 1983), pp. 154-156.

Kear, Fred W. "Board Warp: Causes and Prevention," *Circuits Manufacturing* (December 1983), pp. 95-98.

Lafore, Robert. *Assembly Language Primer for the IBM PC*. Plume/ Waite, San Rafael, California, 1984.

Lancaster, Don. *CMOS Cookbook*. Howard W. Sams & Company, Indianapolis, Indiana, 1977.

Lemons, Phil. "The IBM Personal Computer First Impressions," *Byte* (October 1981), pp. 27-34.

Lemons, Wayne. "Streamlined Tests for Transistors," *Electronic Servicing* (August 1977), pp. 34-39.

Lewis, Gordon. "Disks, Drives, and Dirt," *Pro/Files* (September/October 1983), pp. 59-61.

Lieberman, David. "Data Input Alternatives," *Electronic Products* (June 6, 1983), pp. 47-55.

Lieberman, David. "The Clean Connection," *Nibble*, Vol. 2/No. 8 (1981), pp. 159-165.

Little, M. Andre. "System Security," *inCider* (December 1983), pp. 117-121.

Littlefield, Patti. "What to Try Before Taking Your Microcomputer Into the Repair Department," *Educational Computer Magazine* (May-June 1983), p. 73.

Loop, Roger. "Buying a Digital Scope," *Electronic Products* (May 15, 1987), pp. 38-47.

Machrone, Bill. "How Boca Does It," *PC Magazine* (August 1983), pp. 111-115.

Machrone, Bill. "User-to-User," *PC Magazine* (August 1983), pp. 565-566.

Mann, Timothy J. "Disk Cleaner," *inCider* (October 1983), pp. 166-168.

Margolis, Art. *Troubleshooting & Repairing Personal Computers*. Tab Books, Inc., Blue Ridge Summit, Pennsylvania, 1983.

May, Larry. "Choosing a Keyboard Technology," *Electronic Products* (September 30, 1983), pp. 91-96.

McCain, John. "Spikes: Pesky Voltage Transients and How to Minimize Their Effects," *Byte* (November 1977), pp. 54-56.

McCann, Scott. "Using a Switch-type Joystick on the IBM PC," *PC Tech Journal* (May 1984), p. 195.

McClain, Larry. "Servicing Your System: Be Prepared," *Personal Computing* (September 1982), pp. 50-55, 148-154.

McLanahan, David. "Here are Some Sources for Parts and Information," *Electronic Servicing & Technology* (June 1985), pp. 25-28.

McMullen, Barbare E., and John F. "How Blue Can You Get?" *PC Magazine* (April 3, 1984), pp. 112-113.

Miastkowski, Stan. "A Close Look at the IBM Personal Computer," *Popular Computing* (December 1981), pp. 52-57.

Microsystem Components Handbook: Peripherals Volume II, Intel Corporation, Santa Clara, California, 1986.

Miller, Beth. "Microsystem Reliability Testing," *Electronic Test* (November 1983), pp. 48-54.

Milner, Edward J. "Fast Memory Test Checks Individual Bits," *EDN* (October 13, 1983), pp. 222-229.

Morgan, Chris. "IBM's Personal Computer," *Byte* (July 1981), pp. 6-10.

Morgan, Christopher L. *Bluebook of Assembly Routines for the IBM PC*. New American Library, San Rafael, California, 1984.

Norton, Peter. "Snooping in ROM: The Software Interrupt," *Softalk* (February 1983), pp. 87-89.

Norton, Peter. "Snooping in ROM: The Computer Musician," *Softalk* (March 1983), pp. 79-81.

Norton, Peter. "Snooping in ROM: Which Version Did You Get?" *Softalk* (January 1983), pp. 86-88.

Persson, Conrad. "Oscilloscope: The Eyes of the Technician," *Electronic Servicing & Technology* (April 1987), pp 10-14.

Petzold, Charles. "PC Tutor," *PC Magazine* (August 1986), pp. 439, 442.

Pingry, Julie. "The Expanding Real World of the IBM PC," *Digital Design* (February 1984), pp. 80-88.

Poole, Lon. "Under the Hood of the PC," *PC Magazine* (September 1982), pp. 50-58.

Final Technical Report: RADC Testability Notebook. RADC-TR-82-189, Rome Air Development Center, Griffiss AFB, New York, June 1982.

Radding, Alan. "When Your Computer Breaks Down," *Poplular Computing* (May 1983), pp. 196-198.

Rampil, Ira. "A Floppy Disk Tutorial," *Byte* (December 1977), pp. 24-45.

Rechsteiner, Emil B. "Keeping Power Clean and Steady," *Mini-Micro Systems* (August 1983), pp. 245-252.

Riccio, Ronald. "How to Avoid Damage When Repairing PC Boards," *Electronic Servicing & Technology* (February 1983), pp. 38-42.

Robinson, J. B. *Modern Digital Troubleshooting*. Data I/O Corporation, Redmond, Washington, 1983.

Rosch, Winn L. "High-resolution Color Monitors," *PC Magazine* (June 1983), pp. 247-258.

Sargent, Murry III and Shoemaker, Richard L. *The IBM Personal Computer From the Inside Out*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1984.

Schilling, Robert, Jr. "Hardware Diagnostics for the Home," *Popular Computing* (August 1983), pp. 204-210.

Scovern, John L. "No Corrosion with Antistat," *Circuits Manufacturing* (January 1983), pp. 51-53.

Signetics Logic-TTL Data Manual, Signetics Corp., Sunnyvale, California, 1978.

Sloop, Joe. "Troubleshooting Logic Systems Logically," *Electronic Servicing & Technology* (July 1983), pp. 26-37.

Socha, John. "The Monochrome/Color Switch," *Softalk* (February 1983), pp. 31-33.

Somerson, Paul. "Goblins, Gremlins & Glitches," *PC Magazine* (October 1983), pp. 111-129.

Somerson, Paul. "User-to-User," *PC Magazine* (June 12, 1984), pp. 434-437.

Somerson, Paul. "User-to-User," *PC Magazine* (October 28, 1986), p. 331.

The Primer of High-performance In-circuit Testing, FACTRON, Latham Company, New York, 1985.

Updegraff, Stephen W. "Better Than Gold–Substrate Coating Surpasses Gold in Hi-Rel Connectors," *Circuits Manufacturing* (December 1983), pp. 54-59.

Victor, Jesse. "Low Noise Topologies, Innovative Designs to be Spotlighted at Powercon 9," *EDN* (June 9, 1982), pp. 75-84.

Wattson, Carolyn. "Desoldering Today's Circuit Components," *Electronic Servicing & Technology* (October 1984), pp. 26-28.

Weissman, Ed. "Letters to PC Magazine," *PC Magazine* (August 1986), p. 15.

Whitaker, Lewis A. "Maintenance Alternatives for Personal Computers," *Byte* (June 1982), pp. 452-459.

Williams, Gregg. "A Closer Look at the IBM Personal Computer," *Byte* (January 1982), pp. 36-68.

"Wohl Talks About Peanuts and Other PCs," *Government Computer News* (December 1983), p. 5.

Zachmann, Mark. "PC Tutor," *PC Magazine* (June 1983), pp. 65-83.

Zachmann, Mark. "PC Tutor," *PC Magazine* (July 1983), pp. 69-74.

Zachmann, Mark. "PC Tutor," *PC Magazine* (October 1983), pp. 577-585.

Zachmann, Mark. "PC Tutor," *PC Magazine* (May 29, 1984), pp. 393-397.

Index

R

S

T

U

V

W